

# EVALUACIÓN DE ACCESIBILIDAD EN APLICACIONES NATIVAS ANDROID



Trabajo de Fin de Grado en Ingeniería Informática

Alumno:	Pedro Juan Valle Ten
---------	----------------------

Tutora de proyecto: Ana María Iglesias Maqueda

Fecha de entrega: 27/09/2015

## Resumen

En Internet hay disponibles una gran cantidad de herramientas para comprobar el nivel de accesibilidad de una página web. Sin embargo, no existen apenas herramientas que realicen esa misma función para aplicaciones móviles. Tampoco hay estudios relevantes sobre accesibilidad móvil fuera de los ámbitos de los sistemas operativos para estos terminales (Android, iOS...), por lo que hay un gran vacío en este aspecto.

El primer objetivo de este proyecto es el estudio y evaluación de los diferentes estándares de accesibilidad que hay en la Web, con el fin de ser adaptados a una aplicación móvil en el sistema operativo Android. De esta forma se marcan los requisitos de accesibilidad de toda aplicación móvil nativa.

Como segundo objetivo de este trabajo se ha marcado el de conseguir analizar, diseñar, implementar y verificar una herramienta de evaluación de código de una app de Android. La novedad de esta herramienta consiste en, una vez realizado un pequeño estándar de accesibilidad que se puede aplicar a móviles a partir de los estándares web actuales, que la herramienta desarrollada permita comprobar en el terminal móvil si una app en Android es o no accesible, o qué nivel de accesibilidad tiene.

## Contenido

1. Introducción .....	5
1.1 Planteamiento del problema y motivación.....	5
1.2 Objetivos .....	6
1.3 Alcance .....	6
1.4 Estructura del documento.....	7
2. Gestión y ejecución del proyecto. Análisis de costes.....	9
2.1 Plan de trabajo .....	9
2.2 Metodología de trabajo utilizada .....	9
2.3 Ciclo de vida del proyecto. Descripción de tareas .....	10
2.4 Paquetes de trabajo .....	10
2.5 Planificación inicial y tiempo estimado de realización del trabajo (diagrama de Gantt)..	14
2.6 Planificación final (diagrama de Gantt final).....	16
2.7 Comparativa de planificaciones inicial y final .....	18
2.8 Estimación de horas por fases y tareas.....	18
2.9 Recursos del proyecto .....	20
2.10 Evaluación de costes. Riesgos y oportunidades.....	21
3. Estado del arte .....	23
3.1 Descripción del entorno socio-económico actual .....	23
3.2 Marco regulador, normativas técnicas y legales.....	23
3.3 Trabajos previos, alternativas posibles y trabajos de otros investigadores .....	33
3.4 Herramientas automáticas de evaluación de accesibilidad .....	36
3.5 Discusión .....	42
4. Evaluación y estudio de los estándares y guías de accesibilidad web actuales.....	43
4.1 Propuesta de guía de evaluación de accesibilidad.....	48
4.2 Recomendaciones de usabilidad .....	78
4.3 Conclusiones sobre el estudio de la evaluación de accesibilidad .....	79
5. Requisitos de accesibilidad generales de la aplicación.....	81
6. Desarrollo de una herramienta de evaluación de código accesible .....	95
6.1 Introducción: versión de Android y características de accesibilidad .....	95

6.2 Análisis: requisitos para la app móvil nativa .....	97
6.3 Diseño: interfaz de usuario, funcionalidad, controlador y diagrama de clases .....	106
6.4 Alternativas de diseño.....	109
6.5 Implementación y arquitectura de la aplicación.....	105
6.6 Verificación y pruebas de la herramienta de evaluación.....	109
7. Conclusiones y trabajos futuros .....	115
7.1 Conclusiones, principales aportaciones y consecución de objetivos.....	115
7.2 Dificultades encontradas.....	116
7.3 Trabajos futuros .....	116
8. Glosario de términos.....	118
9. Bibliografía .....	123
9.1 Bibliografía general .....	123
9.2 Bibliografía de las herramientas de evaluación de accesibilidad.....	128
9.3 Bibliografía del desarrollo y la implementación de la aplicación.....	128
Anexo I: Manual de instalación de la aplicación .....	130
Anexo II: Manual de usuario de la aplicación.....	130
Anexo III: Evolución de accesibilidad en Android.....	131



## 1. Introducción

### 1.1 Planteamiento del problema y motivación

En la actualidad las plataformas móviles han alcanzado un grado de importancia enorme para la población, haciendo un uso diario de estos dispositivos, tantos smartphones como tablets o dispositivos similares. El hecho de poder usar en cualquier sitio una aplicación en un terminal móvil, que hace pocos años sólo se podía acceder a ella mediante un ordenador de sobremesa conectado a la red, se ha convertido de una opción con grandes posibilidades para el usuario de estos terminales. Desde el simple acceso a Internet, consulta de redes sociales, aplicaciones ofimáticas u otro tipo de software, se ha posicionado como un gran mercado en constante crecimiento, debido a la variada oferta de dispositivos móviles, así como las diversas aplicaciones que existen para cada plataforma, con sus respectivos mercados de aplicaciones.

Estas aplicaciones van actualizándose continuamente, tanto a nivel de funcionalidad como a nivel de interfaz, y no siempre proporcionan al usuario un nivel de accesibilidad y de usabilidad óptimos.

Entendemos como accesibilidad a la posibilidad de que un producto o servicio web pueda ser usado por el mayor número posible de personas, independientemente de las limitaciones propias del individuo o de las derivadas del contexto de uso. Acceso universal a los recursos independientemente del tipo de hardware, software, infraestructura de red, idioma, cultura, localización geográfica o capacidades de los usuarios.

Y, así mismo, entendemos como usabilidad: Capacidad de un software de ser comprendido, aprendido, usado y ser atractivo para el usuario, en condiciones específicas de uso.

Estos dos conceptos son muy importantes para la satisfacción y confortabilidad de la interacción del usuario con las aplicaciones móviles, y estos aspectos determinan la experiencia de usuario final. Entendiendo como experiencia de usuario el conjunto de factores y elementos relativos a la interacción del usuario, con un entorno o dispositivo concretos, cuyo resultado es la generación de una percepción positiva o negativa de dicho servicio, producto o dispositivo.

Este proyecto tratará de definir con mayor precisión que requisitos de accesibilidad ha de cumplir las aplicaciones móviles nativas en entorno Android, ya que actualmente son incompletas las características generales en accesibilidad móvil, como este estudio pretende demostrar.

Asimismo existe una carencia de herramientas de evaluación de accesibilidad de aplicaciones nativas móviles en las que se profundizará durante el desarrollo de este Trabajo de Fin de Grado, describiendo lo que hay actualmente en el mercado de este tipo de herramientas, y ampliando el estudio de las distintas pautas que ofrecen las guías de accesibilidad, para obtener los requisitos necesarios para el desarrollo de una herramienta de evaluación de accesibilidad, que consiga satisfacer la demanda de este tipo de aplicaciones que tanto escasean en el mercado.

## 1.2 Objetivos

El objetivo principal del proyecto es analizar y evaluar todos los aspectos referentes a la accesibilidad web, para después realizar una comparación con la accesibilidad móvil actual, y conseguir encontrar las carencias de accesibilidad o complementar las mismas o añadir nuevas, a partir del estudio y resumen de una propuesta de accesibilidad, que disponga de los principales aspectos a tener en cuenta en una aplicación móvil. Esta propuesta incluirá los apartados más importantes referidos a accesibilidad de los estándares web más importantes de la actualidad, enfocados al entorno móvil.

Como segundo objetivo, se realizará el análisis, diseño, implementación y verificación de una herramienta que sea capaz de, analizando el código de una aplicación nativa para Android, indicar qué características de accesibilidad están bien implementadas o no dentro de la aplicación móvil, mostrando un resultado similar a la gran cantidad de herramientas web que realizan esta comprobación.

## 1.3 Alcance

El alcance de este proyecto, y por lo tanto, las tareas de las que se compone, son:

- Análisis de las herramientas de evaluación de accesibilidad actuales más importantes.
- Análisis, estudio y comparativa de los estándares de accesibilidad actuales más relevantes para el desarrollo de este proyecto. Se realizará una propuesta que recoja las pautas de accesibilidad necesarias a tener en cuenta para este proyecto.
- Análisis e identificación de los requisitos de la aplicación a partir de las pautas de accesibilidad anteriores.
- Diseño, implementación y verificación de la herramienta que reconoce las pautas o características de accesibilidad de las que dispone una aplicación nativa Android y su carencia de las mismas.
- Elaboración de manuales de instalación y de uso de la aplicación desarrollada.
- Escribir y completar la memoria y posterior defensa.

## 1.4 Estructura del documento

La memoria del proyecto de organizará según estos apartados:

- **Gestión del proyecto:** Incluirá un diagrama de Gantt inicial con una estimación del tiempo que requerirá completar las tareas de cada apartado del proyecto, junto a un diagrama de Gantt final que indicará realmente lo que ha llevado la ejecución de cada tarea.
- **Ejecución del proyecto y análisis de costes:** Se estimará el coste del desarrollo de este proyecto, teniendo en cuenta horas trabajadas en la implementación de la aplicación, si existe alguna licencia de algún lenguaje de programación, entorno de desarrollo, plataforma tecnológica, desarrollo en software de código abierto, etc.
- **Estado del arte:** Se describirá cómo afecta la accesibilidad web y móvil en el entorno socio-económico actual, cuál es su marco regulador y cuales son sus normativas técnicas y legales. Se establecerá una comparación de las distintas herramientas de accesibilidad web que actualmente (trabajos previos) son más relevantes, previo análisis de cada una de las mismas y junto a sus ventajas e inconvenientes.
- **Evaluación y estudio de los estándares de accesibilidad web actuales:** Se realizará un estudio de los estándares de accesibilidad web más importantes, destacando sus puntos fuertes y sus carencias. A partir de ese estudio se elaborará una propuesta de accesibilidad móvil con las pautas más relevantes de cada uno de esos estándares estudiados, y de forma que puedan aplicarse a una plataforma móvil.
- **Requisitos de la aplicación:** Obtención de los requisitos de usuario a partir de las pautas de la propuesta de accesibilidad móvil.
- **Análisis, diseño, arquitectura de la aplicación, implementación y verificación de una herramienta de evaluación:** Se decidirá qué versión de Android utilizar en el desarrollo en función de la versión más usada en la actualidad. Después habrá que reconvertir los requisitos de usuario obtenidos en el apartado anterior, que serían efectivas para aplicaciones web (HTML) **[HTML, Glosario de Términos]** o híbridas (PhoneGap) **[Phonegap, Glosario de Términos]**, en requisitos de usuario para aplicaciones nativas de Android. Se realizará un diseño de clases para el lenguaje de programación Java para a continuación implementar la aplicación en el entorno de programación Eclipse, junto con el plugin ADT para aplicaciones Android. Paralelamente a la implementación se irán realizando pruebas de verificación de la funcionalidad desarrollada.

- **Conclusiones y trabajos futuros:** Se hará una breve descripción de lo conseguido con este proyecto, así como los apartados más complicados de realizar del mismo. Se dejará indicado como continuar con el desarrollo con vistas a la ampliación de este proyecto, de forma que se pueda aumentar las características de accesibilidad móvil de la aplicación.
- **Glosario de términos, bibliografía y anexos:** Estos apartados aportarán información para saber el significado de palabras técnicas, los enlaces bibliográficos utilizados para consultar información en este proyecto. De igual forma se añadirán como anexos dos manuales, uno para la instalación de la aplicación y otro el de usuario, así como un breve esquema de la evolución de accesibilidad en Android desde sus primeras versiones hasta la actualidad.



## 2. Gestión y ejecución del proyecto. Análisis de costes

### 2.1 Plan de trabajo

Se establecerá una planificación inicial a seguir durante el desarrollo del proyecto, con las diferentes etapas por las que irá progresando el mismo.

Inicialmente se hará una breve descripción del entorno y el marco regulador que afecta a la accesibilidad tanto web como móvil.

Se realizará un análisis y comparación de las herramientas actuales más conocidas de accesibilidad web, destacando sus puntos óptimos y mejorables en cuanto a su funcionalidad, para posteriormente ofrecer una visión de lo que puede aportar el estudio realizado en este proyecto.

La parte principal del mismo consistirá en un estudio exhaustivo de los estándares de accesibilidad web actuales, teniendo en cuenta los trabajos previos y la evolución que han ido teniendo estos estándares.

Una vez realizada esta parte, se unificarán los apartados más relevantes de cada uno de ellos realizando una propuesta de accesibilidad para aplicaciones móviles, que contenga las características de accesibilidad que deben tenerse en cuenta en una app móvil [**App Móvil, Glosario de Términos**]. Se justificarán todas las decisiones tomadas, tanto lo que es factible de aplicar al proyecto, como lo que no se contempla.

De esa propuesta de accesibilidad móvil se obtendrán los requisitos funcionales de la aplicación, necesarios para su posterior implementación.

A partir de este punto, se tratará de poner en práctica todas esas características de accesibilidad definidas en los requisitos, mediante el análisis, diseño e implementación de una pequeña aplicación (móvil), que permita descubrir ante el usuario la funcionalidad propuesta, de forma que esta herramienta analice el código de una aplicación móvil y muestre los apartados de accesibilidad de los que carece (también puede indicar los que son óptimos). Se realizarán las pruebas oportunas para verificar el correcto funcionamiento de la herramienta.

Por último, se expondrán las conclusiones a las que ha permitido llegar este proyecto, así como las aportaciones y objetivos que se han conseguido lograr, y que permitirán a futuros trabajos avanzar en esta materia.

### 2.2 Metodología de trabajo utilizada

Se ha optado por utilizar una metodología ágil para el desarrollo del proyecto (como puede ser **SCRUM**). Un desarrollo iterativo e incremental minimiza riesgos y permite desarrollar software en menos tiempo.

Cada iteración del ciclo de vida incluye distintas fases del proyecto como planificación, análisis de requisitos, diseño, implementación, verificación y documentación. Una vez finaliza una iteración, se evalúan de nuevo las prioridades del proyecto.

El poder realizar entregas parciales y regulares facilita la flexibilidad y productividad, y se adapta a requisitos cambiantes, como se puede observar en la Figura 1.



Figura 1: Fases de una Metodología Ágil de desarrollo.

## 2.3 Ciclo de vida del proyecto. Descripción de tareas

Los pasos a seguir indicados en el apartado **1.3 Alcance**, serán desarrollados de forma iterativa e incremental, siguiendo una metodología ágil, de forma que al completar cada iteración, se establecen los siguientes elementos del proyecto a tratar y se revisa lo conseguido hasta la fecha, como se muestra en la Figura 2. Con cada nueva entrega parcial, se modifican las entregas anteriores en casa de necesitar ampliaciones o cubrir alguna carencia detectada, con el fin de optimizar y agilizar el desarrollo.



Figura 2: Pasos iterativos del ciclo de entrega de una Metodología Ágil.

## 2.4 Paquetes de trabajo

Las tareas asociadas con cada fase del proyecto son:

- Introducción:



1. Objetivos: Estudio de estándares web de accesibilidad y análisis, diseño e implementación de la app que analiza si otra app es accesible.
- Gestión:
    1. Planificación.
    2. Ejecución del proyecto y análisis de costes.
    3. Seguimiento: Readaptar la planificación del proyecto cuando sea necesario.
  - Estado del Arte:
    1. Entorno socio-económico. Normativa técnica y legal.
    2. Análisis de herramientas de accesibilidad.
  - Evaluación de estándares de accesibilidad:
    1. Estudio de las pautas de accesibilidad.
    2. Creación de una propuesta de accesibilidad para entorno móvil.
  - Análisis, diseño, implementación y verificación:
    1. Análisis de requisitos de usuario.
    2. Elección de plataforma y arquitectura tecnológica.
    3. Diseño de la interfaz gráfica de aplicación de accesibilidad (vista).
    4. Diseño de la funcionalidad de la aplicación de accesibilidad (modelo).
    5. Diseño para la captura y generación de eventos (controlador).
    6. Alternativas de diseño.
    7. Implementación del código.
    8. Verificación y pruebas.
  - Finalización y presentación del TFG:
    1. Conclusiones.
    2. Completar la memoria.
    3. Elaborar los manuales de instalación y de usuario.

Se puede representar de forma simple el desglose de fases y tareas de todo el proyecto. Para ello se emplea un diagrama WBS o EDT [**Diagrama WBS o EDT, Glosario de Términos**], como se muestra en la Figura 3.

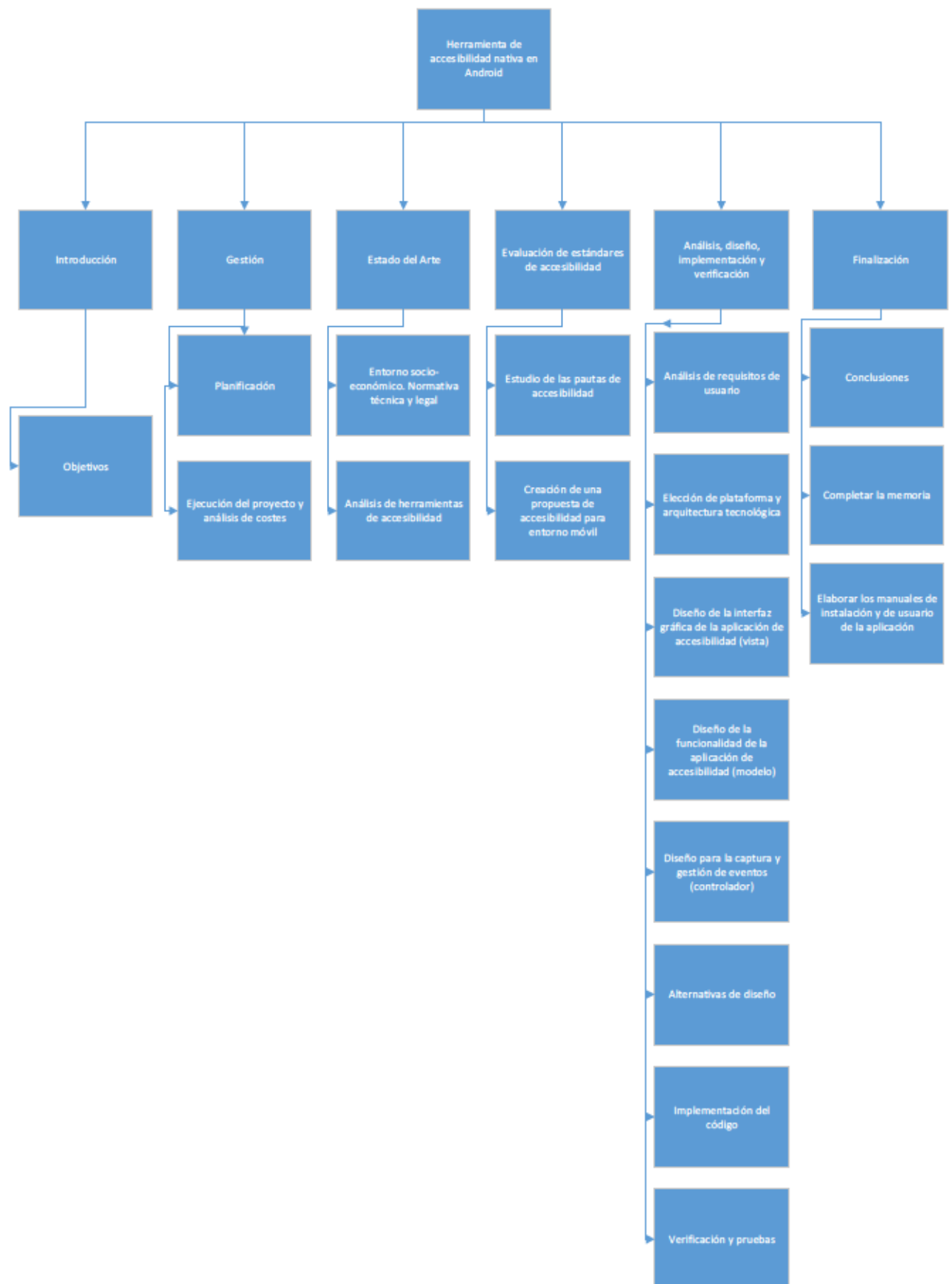


Figura 3: Diagrama de paquetes de trabajo del Trabajo de Fin de Grado.



2.5 Planificación inicial y tiempo estimado de realización del trabajo (diagrama de Gantt)

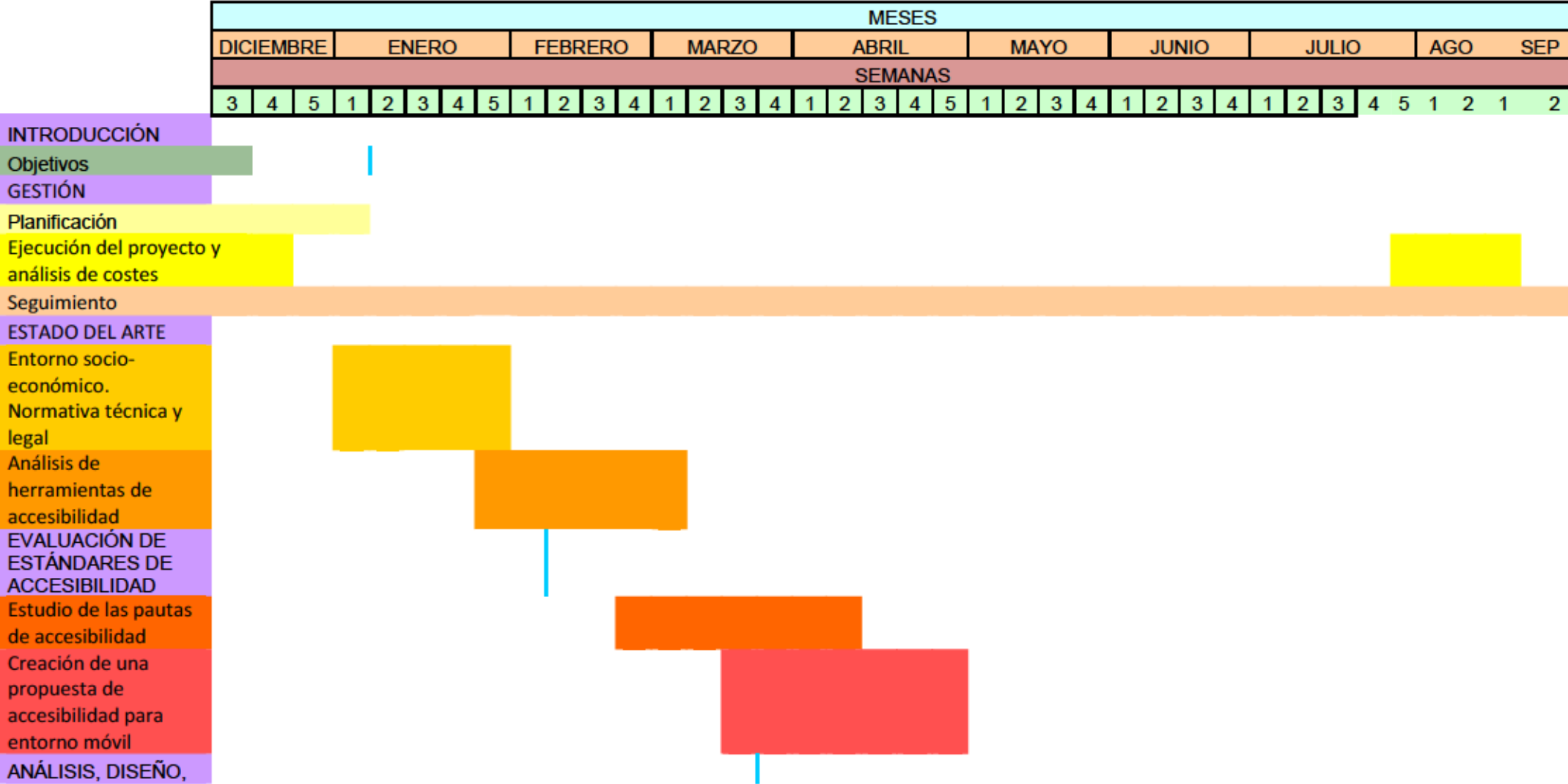
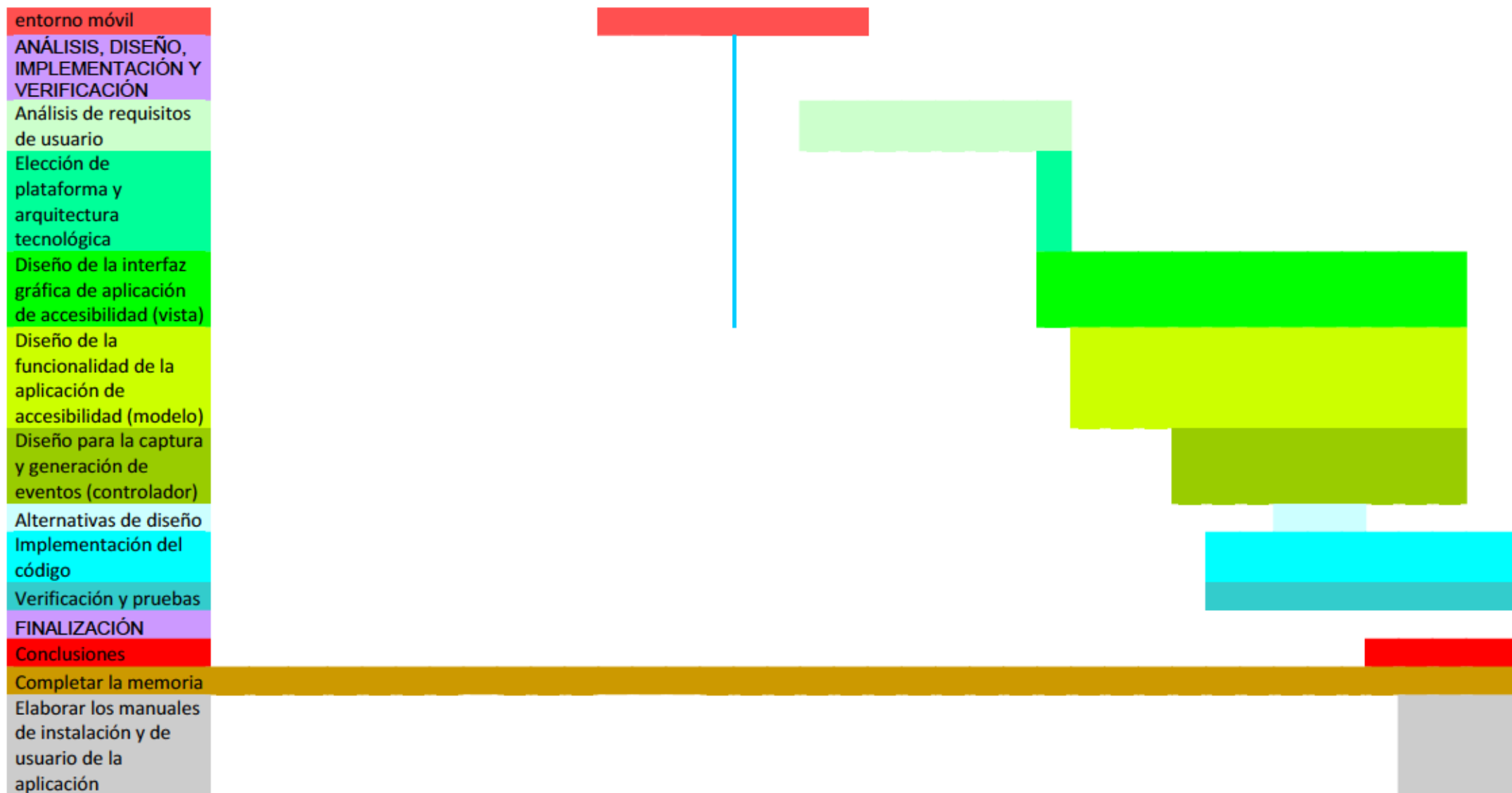




Tabla 1: Diagrama de Gantt inicial.







La tabla 1 muestra la planificación inicial, mientras que la tabla 2 muestra la planificación final.

## 2.7 Comparativa de planificaciones inicial y final

La fecha de inicio del proyecto fue el 22 de diciembre de 2014. Se empezó la planificación inicial del proyecto, que tuvo que ser revisada varias veces, el establecimiento de las fases y tareas del proyecto.

Según avanzaba el mismo, se fue mejorando la planificación, con nuevas fases y tareas, y el tiempo estimado en la planificación inicial para algunas de las tareas se incrementó de manera considerable según queda reflejado en la planificación final.

El seguimiento del proyecto ha facilitado la readaptación del mismo, con el consiguiente aumento de horas dedicadas a diversas tareas que requerían mejoras, nuevas características aportadas, o cubrir algunas carencias, según queda reflejado en la Figura 4.

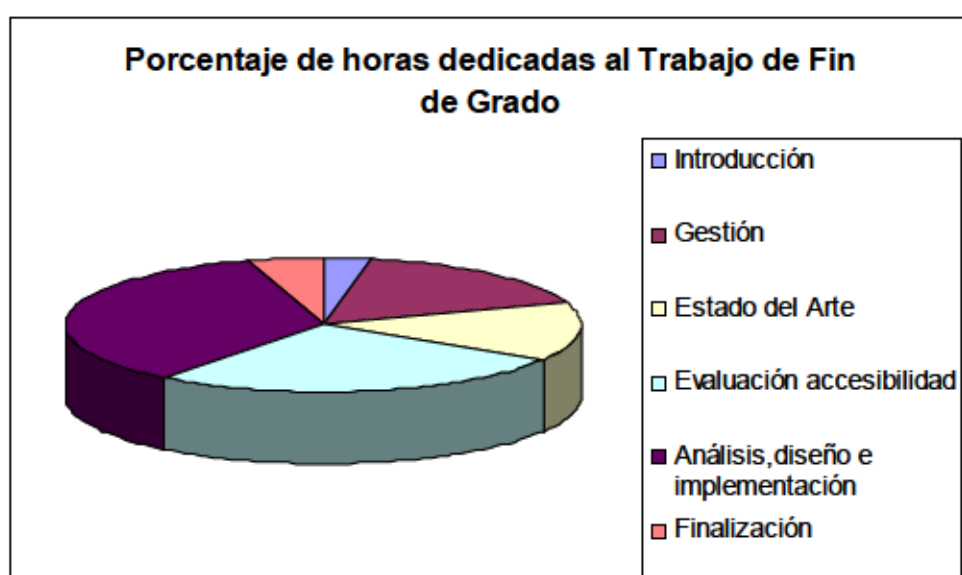


Figura 4: Gráfico circular de la cantidad tiempo empleado en las distintas fases del Trabajo de Fin de Grado.

## 2.8 Estimación de horas por fases y tareas

El trabajo de fin de grado equivale a 12 créditos. Estas horas se repartirán entre las tareas de cada fase del proyecto.

FASE 1: INTRODUCCIÓN	Número de horas estimadas
Tarea 1: Objetivos	6
TOTAL	6

Tabla 3: Estimación de tiempo para la Introducción.

FASE 2: GESTIÓN	Número de horas estimadas
Tarea 1: Planificación	5

<u>Tarea 2:</u> Ejecución del proyecto y análisis de costes	8
<u>Tarea 3:</u> Seguimiento	5
<b>TOTAL</b>	<b>18</b>

Tabla 4: Estimación de tiempo para la Gestión del Proyecto.

<b>FASE 3: ESTADO DEL ARTE</b>	<b>Número de horas estimadas</b>
<u>Tarea 1:</u> Entorno socio-económico. Normativa técnica y legal	25
<u>Tarea 2:</u> Análisis de herramientas de accesibilidad	30
<b>TOTAL</b>	<b>55</b>

Tabla 5: Estimación de tiempo para el Estado del Arte.

<b>FASE 4: EVALUACIÓN DE ESTÁNDARES DE ACCESIBILIDAD</b>	<b>Número de horas estimadas</b>
<u>Tarea 1:</u> Estudio de las pautas de accesibilidad	35
<u>Tarea 2:</u> Creación de una propuesta de accesibilidad para entorno móvil	60
<b>TOTAL</b>	<b>95</b>

Tabla 6: Estimación de tiempo para la Evaluación de Estándares de Accesibilidad.

<b>FASE 5: ANÁLISIS, DISEÑO, IMPLEMENTACIÓN Y VERIFICACIÓN</b>	<b>Número de horas estimadas</b>
<u>Tarea 1:</u> Análisis de requisitos de usuario	20
<u>Tarea 2:</u> Elección de plataforma y arquitectura tecnológica	8
<u>Tarea 3:</u> Diseño de la interfaz gráfica de aplicación de accesibilidad (vista)	6
<u>Tarea 4:</u> Diseño de la funcionalidad de la aplicación de accesibilidad (modelo)	8
<u>Tarea 5:</u> Diseño para la captura y generación de eventos (controlador)	10

<u>Tarea 6:</u> Alternativas de diseño	3
<u>Tarea 7:</u> Implementación del código	35
<u>Tarea 8:</u> Verificación y pruebas	15
<b>TOTAL</b>	<b>105</b>

Tabla 7: Estimación de tiempo para el Análisis, Diseño, Implementación y Verificación de la aplicación.

<b>FASE 6: FINALIZACIÓN</b>	<b>Número de horas estimadas</b>
<u>Tarea 1:</u> Conclusiones	6
<u>Tarea 2:</u> Completar la memoria	10
<u>Tarea 3:</u> Elaborar los manuales de instalación y de usuario de la aplicación	2
<b>TOTAL</b>	<b>18</b>

Tabla 8: Estimación de tiempo para la Finalización del Proyecto.

Total de horas estimadas: 297 horas.

## 2.9 Recursos del proyecto

Los recursos que se necesitan para desarrollar este proyecto son:

### Recursos materiales:

1. Ordenador de sobremesa ACER con 4 GB de RAM, procesador Intel i3-4160 3.6 Ghz o similar, con un tiempo de vida estimado de 8 años, por lo que se irá amortizando cada año. Al año corresponde 65. 63 euros de amortización, y 54.7 euros corresponden a 10 meses de trabajo.
2. Sistema operativo Windows 7 Home Edition.
3. Una línea de ADSL de 6 MB.
4. Programa Microsoft Visio para realizar diagramas como los paquetes de trabajo (diagrama WBS).
5. Programa Visual Paradigm 12.2 para realizar el diagrama UML de clases en Java.
6. Entorno de desarrollo Eclipse 4.4 Luna para la implementación del código de la aplicación.
7. Plugin para Eclipse Android ADT, que incorpore las librerías de desarrollo para la versión de Android elegida (4.4 Kit Kat). Este plugin proporciona un emulador (Android Debug Bridge, "ADB") que ofrece dos máquinas virtuales a elegir para la ejecución de la aplicación (Dalvik o ART), herramientas de compilación y depuración de código.

8. Kit de desarrollo de Java 8, con las librerías del lenguaje.

#### Recursos humanos:

1. Tutora del proyecto, que pertenece a la Universidad donde se presenta el Trabajo de Fin de Grado (TFG). Se encarga de la coordinación y seguimiento del proyecto con el objetivo de resolver dudas y proporcionar feedback para la optimización del desarrollo del proyecto. Siguiendo una metodología ágil, tendría el rol de Product Owner [**Product Owner, Glosario de Términos**], que representa a un cliente.
2. Alumno desarrollador del proyecto de Grado en Ingeniería Informática de la Universidad Carlos III de Madrid (UC3M). El alumno deberá ir adquiriendo conocimientos relativos a accesibilidad web y móvil para poder aplicarlos al desarrollo de la aplicación. Irá progresando en todas las fases y tareas asociadas hasta completar el proyecto. Al seguir una metodología ágil, el alumno tiene el rol de Scrum Master [**Scrum Master, Glosario de Términos**], así como de profesional técnico desarrollador.

## 2.10 Evaluación de costes. Riesgos y oportunidades

Se realizará una estimación de costes del proyecto, como queda reflejado en la Tabla 9..

El desarrollador del proyecto tendrá la categoría de programador junior, y percibirá 25 euros la hora trabajada.

El tutor percibirá 45 euros la hora trabajada.

Recursos	Costes	Horas/Unidades	Coste Total
Tutor	45 euros/hora	35 horas	1575 euros
Desarrollador	25 euros/hora	153 horas	3825 euros
Licencia de Windows 7	240 euros	1 unidad	240 euros
Línea ADSL 6 MB	36 euros/mes	10 meses	360 euros
Ordenador ACER	525 euros	1 unidad	54.7 años euros (amortización a 10 meses)
Microsoft Visio (figuraría como gasto en vez de coste)	0 euros (prueba)	1 unidad	0 euros
Visual Paradigm (figuraría como gasto en vez de coste)	0 euros (prueba)	1 unidad	0 euros
Eclipse 4.4 Luna	0 euros	1 unidad	0 euros

Java JSDK 8	0 euros	1 unidad	0 euros
Android ADT Plugin	0 euros	1 unidad	0 euros
Android SDK 4.4	0 euros	1 unidad	0 euros
			<b>TOTAL: 6054.7 euros</b>

Tabla 9: Costes del proyecto.

El programa Microsoft Visio dispone de una licencia gratuita de un mes para estudiantes universitarios, por lo que no se aplica ningún coste adicional al proyecto.

Se describirán posibles riesgos que tiene este proyecto, y cómo se pueden resolver los mismos.

1. Riesgo: Desbordamiento de información de accesibilidad a analizar y estudiar.

Plan de contingencia: Enfocarse en el entorno móvil para el proyecto.

2. Riesgo: Mala planificación del proyecto.

Plan de contingencia: Readaptar la planificación mediante el seguimiento del proyecto.

3. Riesgo 3: Diseño e implementación de la aplicación con funcionalidad no concretas por dificultad o falta de tiempo.

Plan de contingencia: Centrarse en funcionalidades más concretas, específicas y sencillas, que permitan observar resultados inmediatos. No obsesionarse con complejas gestiones de eventos si es técnicamente muy complejo sincronizarlos.

Las oportunidades que ofrece el desarrollo de este proyecto son:

1. Conocimiento sobre accesibilidad web y móvil y profundización en este campo.
2. Aplicación de accesibilidad al entorno Android, tratando interfaces, librerías y eventos.



### 3. Estado del arte

#### 3.1 Descripción del entorno socio-económico actual

Actualmente, la proliferación de dispositivos móviles va en aumento en la sociedad, permitiendo a los usuarios la búsqueda y selección de información y aplicaciones que deseen utilizar. La conectividad y funcionalidad de estos terminales facilitan el intercambio de información, y a la vez, es un medio de comunicación entre personas (redes sociales, aplicaciones de mensajería, etc.).

La accesibilidad, sea web o móvil, permite que personas con algún tipo de discapacidad puedan entender, interactuar y navegar por la web, a pesar de los diversos problemas o complicaciones que puedan presentar estas personas, sean físicos, auditivos, visuales, neurológicos, del habla o cognitivos. Además, estas limitaciones de acceso las podemos experimentar también personas mayores o personas sin discapacidad de forma temporal. Por ejemplo, al rompernos un brazo -similar a problemas de movilidad física-, en ambientes muy luminosos donde la pantalla del móvil no se ve bien -similar a barreras encontradas por personas con discapacidad visual-, etc.

La accesibilidad también puede mejorar la experiencia de usuario de personas sin discapacidad, optimizando y facilitando su interacción con la aplicación.

A pesar de que algunos dispositivos móviles ya reúnen condiciones para que los usuarios con discapacidad puedan utilizarlos con comodidad, no todas las aplicaciones en el entorno móvil tienen en cuenta las pautas de accesibilidad ni las necesidades que puedan tener los usuarios.

#### 3.2 Marco regulador, normativas técnicas y legales

Las normas de accesibilidad son características que deben cumplir cualquier página web o aplicación en dispositivo móvil, de forma que pueda ser usado por la mayoría de las personas, tanto personas con edad avanzada como con discapacidad, sea con ayudas técnicas o de forma autónoma.

Existen varias leyes, reales decretos, estándares y normas recientes a tener en cuenta a la hora de desarrollar nuevas aplicaciones móviles en España o internacionalmente, así como guías o recomendaciones de accesibilidad no obligatorias que complementan a las pautas de accesibilidad ya establecidas. A continuación, se resumen las más importantes:

**LEYES**

- LEY 11/2007 [LEY 11, 2007], de 22 de junio, de acceso electrónico de los ciudadanos a los Servicios Públicos.

Esta Ley española reconoce el derecho de los ciudadanos a relacionarse electrónicamente con las administraciones públicas, así como la obligación de las mismas a garantizar ese derecho. Pretende fomentar el impulso del uso de los servicios electrónicos en la Administración, incluyendo registros, pagos, notificaciones y la consulta del estado de tramitación de sus procedimientos.

- LEY 27/2007 [LEY 27, 2007], de 23 octubre, por la que se reconocen las lenguas de signos españolas y se regulan los medios de apoyo a la comunicación oral de las personas sordas, con discapacidad auditiva y sordociegas.

Esta Ley española tiene como objetivo la normalización de la lengua de signos española, garantizando su buen uso y contribuyendo a los derechos lingüísticos de las personas que usan esta lengua.

- LEY 56/2007 [LEY 56, 2007], de 28 de diciembre, de Medidas de Impulso de la Sociedad de la Información.

Esta Ley española obliga a las Administraciones Públicas a informar sobre el nivel de accesibilidad de sus páginas web, facilitando un sistema de contacto al usuario para que pueda exponer si encuentra dificultades de acceso al contenido web. Igualmente, las empresas que cumplan una serie de condiciones, deben satisfacer un nivel de accesibilidad en sus páginas web equivalente al nivel de accesibilidad AA del consorcio World Wide Web (W3C). Además, esta Ley aclara que las Administraciones Públicas promoverán la aplicación de los estándares de accesibilidad en todos los elementos basados en las nuevas tecnologías de la Sociedad de la Información.

**REALES DECRETOS**

- REAL DECRETO 366/2007 [REAL DECRETO 366, 2007], de 16 de marzo, de accesibilidad y no discriminación de las personas con discapacidad en sus relaciones con la Administración General del Estado.

Este Real Decreto regula las condiciones de accesibilidad y no discriminación que, respecto de las personas con discapacidad, deben presentar las Oficinas de Atención al

Ciudadano, impresos y cualquier otro medio que la Administración General del Estado dedica específicamente y en el ámbito de sus competencias a las relaciones con los ciudadanos.

- REAL DECRETO 1494/2007 [REAL DECRETO 1494, 2007], de 12 de noviembre, por el que se aprueba el Reglamento sobre las condiciones básicas para el acceso de las personas con discapacidad a la sociedad de la información.

Este Real Decreto obliga a que se cumplan las normas de accesibilidad de las páginas web de la Administración Pública y de muchas empresas. La intención es que todas las empresas adapten sus contenidos web para que sean accesibles.

### RECOMENDACIONES Y GUÍAS DE ACCESIBILIDAD

- Web Content Accessibility Guidelines WCAG 1.0 [WCAG 1.0, 1999]:

Es una recomendación internacional del consorcio World Wide Web (W3C), sobre como hacer accesibles los contenidos de la web a las personas con discapacidad. Se tratará en profundidad en el apartado **4. Evaluación y estudio de los estándares y guías de accesibilidad web actuales**.

- Web Content Accessibility Guidelines WCAG 2.0 [WCAG 2.0, 2008]:

Es una recomendación internacional del consorcio W3C sobre cómo hay que hacer accesibles los contenidos web a personas con discapacidad que fueron aprobados en diciembre de 2008. Tiene en cuenta las nuevas tecnologías y toma como base el uso de las WCAG 1.0. Se estudiará con detenimiento en el apartado **4. Evaluación y estudio de los estándares y guías de accesibilidad web actuales**.

- WAI-ARIA Accessible Rich Internet Applications [WAI-ARIA 1.1, 2014]:

WAI-ARIA fue aprobada en marzo de 2014, es una recomendación internacional que define cómo hacer aplicaciones y contenido web más accesible para personas con discapacidad. Está especialmente enfocado para la generación de contenido dinámico y para interfaces de usuario avanzadas desarrolladas con AJAX, HTML y Javascript. Esta englobada en la Iniciativa para la Accesibilidad Web (WAI) del Consorcio de la Web (W3C). Se profundizará en el apartado **4. Evaluación y estudio de los estándares y guías de accesibilidad web actuales**.

### METODOLOGÍAS

- Metodología Unificada de Evaluación Web UWEM 1.0 [UWEM 1.0, 2006]:

Describe una metodología de evaluación del cumplimiento de la recomendación WCAG 1.0. El objetivo de esta metodología es asegurar que las herramientas y técnicas de evaluación desarrolladas para la vigilancia a gran escala o la evaluación en local, sean coherentes y compatibles entre sí y con la iniciativa WAI. La próxima versión de UWEM tendrá en cuenta la migración de las WCAG 1.0 a WCAG 2.0. Abarca un sistema de principios y de prácticas para la evaluación de la accesibilidad de la web tanto por una persona experta como de forma automática por interfaces de máquinas. Esta metodología se puede usar para evaluaciones detalladas de una páginas web, un sitio entero o múltiples sitios.

## ESTÁNDARES DE ACCESIBILIDAD

- Estándar ISO/IEC 40500:2012 [ISO/IEC 40500, 2012]:

Se aprobó en octubre de 2012. Es un estándar internacional que recoge las pautas de accesibilidad web WCAG 2.0, y que ayuda a la armonización de los niveles de accesibilidad web entre países y aumenta las posibilidades de adopción de la tecnología y las directrices del W3C.

## NORMAS

- Norma UNE 139803:2004 [UNE 139803, 2004]. Requisitos de Accesibilidad para contenidos en la web:

Es una norma española que toma como punto de partida las pautas de accesibilidad de la WCAG 1.0. La mayoría de los requisitos de accesibilidad son equivalentes en ambas pero la norma UNE es algo más exigente y algunos de sus requisitos de accesibilidad tienen mayor prioridad. Una web que cumpla la norma UNE cumplirá también las pautas de accesibilidad WCAG 1.0 pero no al revés, ya que se tendrá que realizar alguna modificación para alcanzar un determinado nivel de accesibilidad según la norma UNE.

La legislación española establece mediante el REAL DECRETO 1494/2007 de 12 de noviembre, la aprobación del Reglamento sobre las condiciones básicas para el acceso de las personas con discapacidad a la sociedad de la información, con el que se obliga a las webs de las Administraciones Públicas a cumplir los requisitos de prioridad 1 y 2 especificados en la norma UNE.



- Norma CWA 15554:2006 [CWA 15554, 2006]. Especificaciones para el esquema de la evaluación de la conformidad y marca de calidad sobre accesibilidad web:

Es una norma europea avalada por el Comité Europeo de Normalización (CEN), establecida con fecha de 15 de abril de 2006. Constituye la base de la certificación europea en Accesibilidad Web y refleja tres tipos de peticiones que fueron solicitadas: declaración de conformidad por proveedores, inspección y certificación de productos. Igualmente fomenta la mejora y armonización de esquemas de etiquetado locales. Con esta norma se estableció la creación de un Instituto Central Europeo que organiza y controla las prácticas en colaboración con los cuerpos internacionales implicados.

Las actividades propuestas por este instituto para esta norma son: evitar la fragmentación de las marcas de calidad sobre accesibilidad web y desarrollo de la misma, especificar los criterios sobre accesibilidad web en línea con la versión adoptada de las WCAG de WAI, establecimiento de acuerdos con las partes que otorguen la marca de calidad sobre accesibilidad web y registro de los sitios web que hayan obtenido esa marca.

Esta norma esta en consonancia con la Metodología Unificada de Evaluación Web UWEM 1.0.

- Norma UNE 139803:2012 [UNE 139803, 2012]. Requisitos de Accesibilidad para contenidos en la web:

Esta norma española aprobada en julio de 2012 establece los requisitos de accesibilidad para los contenidos web. Tiene en cuenta las Pautas de Accesibilidad para el contenido web WCAG 2.0 de la Iniciativa para la Accesibilidad Web del consorcio de la web (W3C). Es una actualización de la norma UNE-139803:2004.

- UNE-EN 301 549 [UNE-EN 301 549, 2014], "Requisitos de accesibilidad de productos y servicios TIC aplicables a la contratación pública en Europa" - "Accessibility requirements suitable for public procurement of ITC products and services in Europe":

Establece los requisitos funcionales que asegura que los productos y servicios sean accesibles para todas las personas, sea un dispositivo móvil, un ordenador de sobremesa o páginas web. Los requisitos para la web se basan en las pautas de accesibilidad de contenidos web WCAG 2.0, de forma similar a los indicados en la norma española UNE 139803 "Requisitos de Accesibilidad para contenidos en la Web".

Esta norma europea describe los procedimientos de ensayo y la metodología de evaluación de cada uno de ellos, y será adoptada en España como UNE-EN 301 549 por AENOR, que es la entidad legalmente responsable del desarrollo de las normas técnicas en España.

Asimismo existen instituciones oficiales y asociaciones que se encargan de facilitar el acceso a la información y los servicios, de forma simple, con facilidad y sin restricciones.

### INSTITUCIONES OFICIALES

- AENOR (Asociación Española de Normalización y Certificación) [AENOR, 1986], es la entidad legalmente responsable del desarrollo de las normas técnicas en España, creada en 1986 y sin fines lucrativos. Contribuye a mejorar la calidad y competitividad de las empresas, sus productos y servicios, diseña software para la gestión de sistemas y ofrece distintos servicios de información.
- W3C (World Wide Web) [W3C, 1994], es una comunidad internacional donde organizaciones y personas trabajan conjuntamente para desarrollar estándares web.

### ASOCIACIONES

- SIDAR (Seminario de Iniciativas sobre Discapacidad y Accesibilidad en la Red) [SIDAR,2002], es una asociación cuyos fines informativos y formativos, investigación y desarrollo y de asesoría. Sigue las indicaciones que dicta la WAI y sus pautas de accesibilidad mínimas.
- DISCAPNET [DISCAPNET, 2014], es una asociación que fomenta la transmisión de información de una persona a otra, adaptándose a diversos tipos de comunicaciones digitales mediante audiodescripción, subtítulos y lenguas barille de signos.

A continuación, a modo de resumen, se muestra en la Tabla 10 todas las leyes, estándares, guías e instituciones descritas anteriormente y cómo se relacionan entre sí.





LEYES	REALES DECRETOS	GUÍAS	METODOLOGÍAS	ESTÁNDARES	NORMAS	INSTITUCIONES	ASOCIACIONES
<p>LEY 11/2007</p> <p>Acceso electrónico de los ciudadanos a los Servicios Públicos</p>	<p>REAL DECRETO 366/2007</p> <p>Accesibilidad y no discriminación de las personas con discapacidad en sus relaciones con la Administración General del Estado</p>	<p>WCAG 1.0</p> <p>Cómo hacer accesibles los contenidos de la web a las personas con discapacidad</p>	<p>UWEM 1.0</p> <p>Evaluación del cumplimiento de la recomendación WCAG 1.0</p>	<p>ISO/IEC 40500:2012</p> <p>Recoge las pautas de accesibilidad web WCAG 2.0</p>	<p>UNE 139803:2004</p> <p>Requisitos de Accesibilidad para contenidos en la web basado en WCAG 1.0</p>	<p>AENOR</p> <p>Asociación Española de Normalización y Certificación</p>	<p>SIDAR</p> <p>Seminario de Iniciativas sobre Discapacidad y Accesibilidad en la Red</p>
<p>LEY 27/2007</p> <p>Reconocimiento de las lenguas d signos españolas para personas sordas o con</p>	<p>REAL DECRETO 1494/2007 Acceso de las personas con discapacidad a la sociedad de la información</p>	<p>WCAG 2.0</p> <p>Amplía a WCAG 1.0.</p>			<p>Norma CWA 15554:2006</p> <p>Especificaciones para el esquema de la evaluación de la conformidad y</p>	<p>W3C</p> <p>Comunidad internacional donde organizaciones y personas trabajan conjuntamente para desarrollar estándares</p>	<p>DISCAPNET</p> <p>Asociación que fomenta la transmisión de información de una persona a otra</p>

discapacidad auditiva					marca de calidad sobre accesibilidad web	web	
LEY 56/2007  Medidas de Impulso de la Sociedad de la Información		WAI-ARIA 1.1  Cómo hacer aplicaciones y contenido web más accesible para personas con discapacidad orientado e contenido dinámico e interfaces de usuario avanzadas			Norma UNE 139803:2012  Requisitos de Accesibilidad para contenidos en la web basado en WCAG 2.0		
					UNE-EN 301 549  Requisitos de accesibilidad de		

					productos y servicios TIC aplicables a la contratación pública en Europa		
--	--	--	--	--	--------------------------------------------------------------------------	--	--

Tabla 10: Leyes, reales decretos, guía, metodologías, estándares, normas, instituciones y asociaciones referentes a accesibilidad.

### 3.3 Trabajos previos, alternativas posibles y trabajos de otros investigadores

En el apartado 4 se estudiarán en profundidad algunos de los estándares y recomendaciones de accesibilidad más importantes para realizar una propuesta de guía de accesibilidad que sirva para evaluar aplicaciones móviles.

Existen diferentes empresas, grupos y organizaciones que han desarrollado sus propias guías de recomendación sobre las pautas que todo software o aplicación web debería cumplir para ser accesible. A continuación, se enumeran brevemente las más complejas y de las que se parte en este proyecto.

- El **consorcio World Wide Web (W3C)** [W3C, 1994], promueve el desarrollo de estándares web, con el trabajo de organizaciones y personas de forma conjunta. A través de su iniciativa (WAI) [**WAI, Glosario de Términos**] (Web Accessibility Initiative), desarrolla guías y recomendaciones referentes a los diferentes aspectos de la accesibilidad Web. Para los contenidos Web ha creado las Pautas de Accesibilidad de Contenido Web 2.0 [WCAG 2.0, "<http://www.sidar.org/traducciones/wcag20/es/>"], basadas en las WCAG 1.0 [WCAG 1.0, 1999]. Hoy en día, las WCAG 2.0 se han convertido en un estándar internacional cuyas pautas cubren un amplio rango de recomendaciones para la creación de contenido web más accesible. Permite a un mayor número de personas con diversa discapacidad (ceguera y baja visión, sordera y deficiencias auditivas, deficiencias del aprendizaje, limitaciones cognitivas y de movilidad, deficiencias del habla y fotosensibilidad), el seguimiento de estas pautas ayuda a que el contenido web sea más accesible y usable para cualquier persona. Igualmente el consorcio W3C, dispone de una guía que especifica las Mejores Prácticas para mejorar la experiencia de usuario al acceder a dispositivos móviles, MWBP [Mobile Web Best Practices 1.0, "<http://www.w3.org/TR/mobile-bp/>"], así como una ayuda al desarrollo de aplicaciones web móviles para que el contenido sea más rico y dinámico, en el que tiene en cuenta las pautas de ingeniería más relevantes, enfocándose en las que permiten una mejor experiencia de usuario y la advertencia de las pautas que se consideran perjudiciales, MWABP [Mobile Web Application Best Practices, "<http://www.w3.org/TR/mwabp/>"].
- **Funka Nu** [Funka Nu, 2000] es una organización de Suecia que fue desarrollando como proyecto unas pautas para el desarrollo de interfaces móviles accesibles, dependiente de la Fundación para Infraestructuras de Internet. Definieron una metodología en colaboración con el movimiento asociativo de las personas con discapacidad. Las guías desarrolladas por Funka Nu [Funka Nu Guidelines, "<http://www.funkanu.com/en/Our-Expertise/Research-projects/Arkiv-Forskningsrapporter-och-fortroendeuppdrag/Mobile-accessibility-guidelines1/>"] están basadas en las pautas internacionales del estándar de accesibilidad a los contenidos web (WCAG 2.0). Han realizado muchas pruebas con usuarios con distintas necesidades y condiciones, demostrando que las pautas WCAG 2.0 no son suficientes para el entorno móvil, de ahí



el enfoque que este proyecto quiere ofrecer para mejorar la accesibilidad en dispositivos táctiles.

- **BBC** [BBC, 2005] es una organización que incorpora una serie de estándares promovidos por la W3C para facilitar el acceso a los contenidos publicados en sitios web a aquellos usuarios que utilizan dispositivos especiales, como lectores de pantalla. Dispone de una guía de buenas prácticas de accesibilidad [BBC, "<http://www.bbc.co.uk/guidelines/futuremedia/accessibility/mobile>"] para Internet móvil. Estas prácticas tienen en cuenta contenido web, aplicaciones híbridas y nativas para dispositivos móviles. Está muy enfocado en la experiencia de usuario.
- **DEQUE SYSTEMS** [DEQUE SYSTEMS, 1999], es una empresa que contribuye a que las personas con discapacidad puedan disfrutar plenamente de la web. Han desarrollado la herramienta de evaluación de accesibilidad Deque System Axe [Deque System Axe, "<http://www.deque.com/products/axe/>"], que se explicará y se tratará en el apartado **3.4 Herramientas automáticas de evaluación de accesibilidad**
- **SSB BART Group** [SSB BART Group, 1997] es una empresa que apoya las iniciativas de accesibilidad de muchas organizaciones mediante el desarrollo de software y soluciones de consultoría, de forma que afronta las necesidades del negocio y técnicas específicas de las organizaciones más afectadas por los requisitos de accesibilidad. Han creado la Plataforma de Gestión de Accesibilidad (AMP), un software que proporciona la infraestructura para gestionar la accesibilidad en todo el ciclo de vida de desarrollo de la empresa [SSB BART Group, "<http://www.ssbbartgroup.com/>"]. Recientemente han desarrollado esa plataforma enfocada a entorno móviles nativos (Android e iOS), AMP Mobile [AMP Mobile, "<http://www.ssbbartgroup.com/blog/amp-for-mobile-is-here/>"], de la que se profundizará en el apartado **3.4 Herramientas automáticas de evaluación de accesibilidad**.
- Por último, destacar a **Paul J. Adam** [Paul J. Adam], consultor de accesibilidad en el mundo empresarial, que propone una serie de pautas para accesibilidad móvil y web, enfocándose en servicios, presentaciones, demos, aplicaciones y blogs [Paul J. Adam, "<http://pauljadam.com/favelets/>"].

De las guías de accesibilidad más importantes aquí nombradas se tratarán en profundidad en el apartado **4. Evaluación y estudio de los estándares y guías de accesibilidad web actuales**.

A continuación se muestra en la Tabla 11, un breve resumen de las empresas u organizaciones y guías o software de accesibilidad.








		ORIENTADO A WEB MÓVIL	ORIENTADO A APP. MÓVIL NATIVA
W3C – WAI	WCAG 2.0		
	MWBP		
	MWABP		
Funka Nu	Funka Nu Guidelines		
BBC	Mobile Accessibility Guidelines		
DEQUE SYSTEMS	Deque System Axe		
SSB BART Group	AMP Mobile		

Tabla 11: Empresas y organizaciones y sus respectivas guías o software de accesibilidad.

Para el desarrollo de este proyecto se tendrá en cuenta tanto las guías de accesibilidad como el software que algunas empresas puedan ofrecer para comparar resultados con la aplicación desarrollada.

Por un lado las soluciones software de accesibilidad Deque System Axe y AMP Mobile de sus respectivas empresas, serán analizadas en el siguiente apartado **3.4 Herramientas automáticas de evaluación de accesibilidad**, con especial atención a la herramienta AMP Mobile, que propone la evaluación de accesibilidad en aplicaciones nativas en entornos Android e iOS, siendo en el caso de Android, el mismo objetivo a conseguir en este proyecto, pero aplicando una visión distinta del mismo, como veremos más adelante.

En cuanto a las recomendaciones y guías de accesibilidad que ofrecen las organizaciones, el consorcio W3C dispone de unas guías de accesibilidad más generales y más enfocadas a la web que al entorno móvil. Las pautas de accesibilidad WCAG 2.0 no se adaptan demasiado bien a dispositivos móviles, están incompletas, y por eso este proyecto toma como base las guías de accesibilidad que ofrece la empresa Funka Nu, muy bien enfocadas a dispositivos móviles en casi todas las características de accesibilidad. Como complemento a éstas y orientadas a la experiencia de usuario, las guías de accesibilidad que ofrece la organización BBC sirven también de ayuda para conseguir la propuesta de guía de accesibilidad en entornos móviles

que se realizará en el apartado **4. Evaluación y estudio de los estándares y guías de accesibilidad web actuales**.

En menor medida pero también analizadas exhaustivamente en el apartado 4 de este proyecto, se estudiarán las recomendaciones para realizar una implementación accesible MWBP y las recomendaciones de accesibilidad enfocadas a la experiencia de usuario MWABP, que también ofrece el consorcio W3C.

Por último, el enfoque que aporta el consultor experto en accesibilidad. Paul J. Adam, se tendrá en cuenta igualmente, más enfocado para una futura implementación.

### 3.4 Herramientas automáticas de evaluación de accesibilidad

Existen diferentes herramientas automáticas o semi-automáticas de evaluación de accesibilidad, con diferencias significativas entre unas y otras. A continuación, se mencionan algunas de las herramientas web actuales más importantes, observando sus puntos fuertes y débiles, cómo se puede ver en la lista de las herramientas de evaluación de accesibilidad del consorcio W3C [Web Accessibility Evaluation Tools List, "<http://www.w3.org/WAI/ER/tools/>"].

- 1) La herramienta **WAVE** [WAVE, "<http://wave.webaim.org/>"] permite evaluar la accesibilidad de una página web, ya sea indicando la URL de la página a analizar o subiendo directamente la página. Permite elegir el nivel de análisis, tiene bastantes opciones de configuración y proporciona mucha información en los informes de análisis.

Las opciones de configuración están agrupadas en dos conjuntos:

- a. **"Document mode"**, permite seleccionar qué elementos de la página original se tienen que mostrar en el informe, sean tablas, imágenes, multimedia...
- b. **"WAVE features to include"**, permite indicar que características de análisis se tienen que incluir en el informe, sean errores, alertas o información estructural sobre el documento.

WAVE tiene algunas ventajas sobre el resto de herramientas de accesibilidad, como la detección de si se está empleando algunas características para mejorar la accesibilidad de una página o si desaconseja su uso. Para los enlaces invisibles en la misma página aconseja hacerlos visibles para los usuarios con discapacidad motora. Para los atajos de teclado desaconseja su uso debido a posibles conflictos con los atajos de otros programas (como los atajos del propio navegador). En cuanto al orden de tabulación indicado desaconseja su uso debido a irregularidades en su implementación. Por último, los iconos que se muestran en los informes de análisis se explican con detalle, de forma que al hacer una prueba con



una página web, se observa como aparecen alertas de muchas de las reproducciones de vídeo contenidas en la página.

- 2) La herramienta de accesibilidad web **HERA** [HERA, "<http://www.sidar.org/hera/>"] realiza un análisis automático preliminar y da soporte para realizar una revisión manual de las páginas web que se están visualizando en el navegador web. Indica si se encuentran errores, que son detectados en forma automática y qué puntos de verificación de las pautas deben ser revisados manualmente. Esta revisión manual es imprescindible para comprobar si la página es accesible, y sigue las pautas de accesibilidad para contenido web WCAG 1.0.

Para facilitar la revisión manual, se ofrece información de los elementos a verificar, instrucciones sobre cómo realizar ese control y dos vistas modificadas de la página (una en modo gráfico, otra del código HTML) con los elementos más importantes destacados con iconos y distintos colores. Un formulario permite modificar los resultados automáticos, agregar comentarios a cada punto de verificación e indicar el nombre del revisor y se puede generar el informe final sobre la revisión en distintos formatos (XHTML, RDF y PDF).

- 3) La herramienta de accesibilidad web **TAW** [TAW, "<http://www.tawdis.net/>"] dispone de analizador para páginas webs en una pestaña y en otra ha comenzado con una beta para ofrecer un resumen de las carencias de accesibilidad de una aplicación móvil basada en web o híbrida. Dispone de dos modalidades de ejecución, ya sea como aplicación instalable de forma local o como analizador online desde su página web.

Una vez introducida la URL de la página a analizar en la herramienta de accesibilidad, se ofrece un resumen generando un informe en HTML que indica el nombre de la web analizada, en que pautas de accesibilidad se ha basado el análisis de la web (las guías de accesibilidad WCAG 1.0 [WCAG 1.0, 1999] y WCAG 2.0 [WCAG 2.0, 2008], y la Norma UNE 139803 [UNE 139803, 2004], así como las tecnologías que se han tenido en cuenta para el mismo (HTML, CSS). Las distintas pautas de accesibilidad están agrupadas en función de si cumplen una serie de comprobaciones o criterios de éxito en 3 niveles de accesibilidad (A, AA, AAA), siendo el nivel del análisis de esta herramienta AA por defecto para la versión online.

Sin embargo, el nivel de accesibilidad que usa la herramienta para la versión instalable de forma local es sólo A, pero a cambio, incorpora características adicionales como análisis multipágina para realizar un análisis exhaustivo del sitio web, un informe resumen de accesibilidad de las páginas web analizadas, revisión manual de incidencias con inclusión de

comentarios y con la posibilidad de modificar su validez, creación de verificaciones personalizadas y configuración individual de los de verificación que se van a analizar.

Ambas herramientas (online e instalable en local) describen por separado los problemas, advertencias e incidencias no verificadas de forma muy clara, clasificando cada uno de los inconvenientes encontrados en su respectiva categoría de pautas de accesibilidad de WCAG 2.0. Al encontrar problemas indica claramente que son necesarias correcciones y en cuanto a las advertencias, hay que revisarlas manualmente cada una de ellas, al igual que las incidencias no verificadas en el análisis.

La herramienta dispone de un enlace a un informe detallado de cada uno de los apartados que necesitan revisión, indicando el tipo de pauta de accesibilidad en la que está englobado el problema detectado, así como una pequeña descripción del mismo, y el número de línea en el que se encuentra. Además, se indican con los diferentes iconos (rojo para correcciones obligatorias y amarillo para revisión manual) los problemas detectados sobre el código fuente la página web.

- 4) **DEQUE SYSTEM AXE** (THE ACCESSIBILITY ENGINE) [DEQUE SYSTEM AXE, "<http://www.deque.com/products/axe/>"] es una nueva herramienta de accesibilidad web publicada recientemente que realiza una prueba de accesibilidad automatizada, teniendo en cuenta el navegador y el tipo de pruebas a realizar. Se ejecuta en el servidor de desarrollo local en el mismo navegador, de forma que su ejecución hace la prueba de accesibilidad rápida, confidencial y segura, a diferencia de otros servicios web. Se descarga un módulo para Javascript (con funciones para llamadas a páginas a comprobar su accesibilidad), aunque hay que tener instalado Node.js.
- 5) **eXaminator** [eXaminator, "<http://examinator.ws/>"] es un servicio online para evaluar de forma automática la accesibilidad de una página web, y tiene en cuenta las Pautas de Accesibilidad para el Contenido Web 2.0 [WCAG 2.0]. Adjudica una puntuación de 1 a 10 como indicador rápido del nivel de accesibilidad de la página web y ofrece un informe detallado de las pruebas realizadas. La revisión automática no satisface todos los criterios de conformidad (niveles de accesibilidad en función de los criterios comprobados, A, AA, AAA).
- 6) **AMP MOBILE** [AMP MOBILE, "<http://www.ssbartgroup.com/blog/amp-for-mobile-is-here/>"] es una Plataforma de Gestión de Accesibilidad para aplicaciones nativas móviles y también tiene una versión para contenido web. Es un nuevo motor de pruebas para el



entorno móvil en el que se realiza una prueba automatizada de la accesibilidad de dispositivos móviles directamente desde los entornos de desarrollo móvil y de forma ágil. Identifica y resuelve violaciones de accesibilidad a través de pruebas rápidas de forma iterativa en las primeras etapas de desarrollo, en lugar de hacer un simple análisis de código fuente. Las pruebas se pueden realizar en el mismo dispositivo móvil también.

Esta herramienta también es compatible con aplicaciones nativas Android e iOS, disponiendo en cada plataforma de una librería de accesibilidad, junto a unas recomendaciones de buenas prácticas. Ofrece de forma automatizada y organizada acceso e información de los nuevos estándares de accesibilidad que se van incorporando a la herramienta.

Las ventajas de las herramientas de evaluación de accesibilidad que merecen ser destacadas son:

1. WAVE destaca al indicar visualmente las correcciones a realizar tras el análisis y las advertencias en una vista de la página web analizada, junto con los iconos correspondientes. También realiza la detección teniendo en cuenta si se están empleando algunas características para mejorar la accesibilidad de una página o si desaconseja su uso. También permite elegir qué elementos de la página web analizar y qué características del análisis, como errores o advertencias, deben incluirse al generar el informe final.
2. HERA ofrece un gran soporte para realizar una revisión manual de las páginas web y permite generar el informe final de la revisión en distintos formatos.
3. TAW es la herramienta de evaluación que más información detallada ofrece al usuario, gracias al amplio desglose de su análisis, indicando claramente cada una de las correcciones a realizar y de las advertencias que deben comprobarse manualmente.
4. DEQUE SYSTEM AXE ofrece más seguridad y confidencialidad que otras herramientas de accesibilidad.
5. eXaminator otorga un indicador rápido del nivel de accesibilidad de la página web con un número del 1 al 10.
6. AMP MOBILE ofrece una herramienta de accesibilidad para aplicaciones nativas Android. Las demás herramientas de accesibilidad sirven para contenido móvil basado en web o aplicaciones híbridas (como PhoneGap). Identifica y resuelve violaciones de accesibilidad a través de pruebas rápidas de forma iterativa en las primeras etapas de desarrollo.

Los inconvenientes las herramientas de evaluación de accesibilidad que son más relevantes son:

1. WAVE muestra excesivas advertencias si existen atajos de teclado en la página web.
2. HERA no es una herramienta de accesibilidad tan intuitiva para el usuario como lo son TAW y WAVE.
3. TAW al ser utilizado en forma de instalable local sólo ofrece un criterio de conformidad de nivel A al aplicar las pautas WCAG 2.0 en la evaluación de accesibilidad de la página web.
4. DEQUE SYSTEM AXE es más compleja de usar para un usuario sin conocimientos informáticos al tener que descargar un módulo para Javascript.
5. eXaminator sólo permite revisar un número limitado de páginas web por sesión.
6. AMP MOBILE no realiza un análisis de todo el código fuente de la aplicación móvil nativa, por lo que al generar el informe final de evaluación de accesibilidad podría no mostrar todos los errores y advertencias que contiene la aplicación.

A continuación se realiza un breve resumen, que se muestra en la Tabla 12, indicando para cada herramienta de evaluación de accesibilidad, si está orientada a la web o a dispositivos móviles, así como sus ventajas y sus desventajas.

Tabla 12: Herramientas automáticas de evaluación de accesibilidad.

	WAVE	HERA	TAW	DEQUE SYSTEM AXE	eXaminator	AMP MOBILE
Web/Móvil	Web	Web	Web	Web	Web	Móvil
Ventajas	Informe final gráfico, elección de elementos y características a analizar	Informe final en distintos formatos, soporte para revisión manual	Informe final con abundante información detallada	Seguridad y confidencialidad	Indicador rápido del nivel de accesibilidad de la web	Analiza aplicaciones nativas Android e iOS mediante pruebas rápidas
Desventajas	Exceso de advertencias para atajos de teclado	No es demasiado intuitiva	Limita los criterios de conformidad a un nivel A al instalarse localmente	Compleja de usar	Análisis limitado a un número de páginas web por sesión	No realiza un análisis de todo el código fuente de la aplicación

### 3.5 Discusión

Existen numerosas guías y recomendaciones de accesibilidad para aplicaciones web, que ofrecen una serie de principios y de pautas a seguir para cumplir con los criterios mínimos de accesibilidad de la aplicación.

Sin embargo, hoy por hoy, en plataformas móviles no existen aún unas guías o pautas claras y completas a seguir para poder desarrollar una aplicación móvil nativa que sea accesible.

Por ello, el objetivo número de este Trabajo Fin de Grado será el de realizar una recomendación de pautas y guías para el desarrollo y evaluación de aplicaciones móviles nativas, específicas para entornos móviles. Esta recomendación estará basada en las pautas y guías más importantes de accesibilidad que se han estudiado en el estado del arte, así como los principales desarrollos de evaluación de accesibilidad que se han realizado sobre aplicaciones móviles (tanto Web como nativas).

Así mismo, también se ha observado la falta de herramientas automáticas de evaluación de accesibilidad de aplicaciones nativas móviles, por lo que el segundo objetivo de este proyecto es identificar los requisitos de una aplicación que de evaluación de accesibilidad que servirá para el entorno web, híbrido o nativo, basándonos en la propuesta de guía realizada. Posteriormente se realizará el desarrollo completo de un prototipo de aplicación de evaluación automática y las pruebas del mismo.

Recientemente ha salido al mercado el software para dispositivos móviles AMP Mobile, que analiza la accesibilidad para aplicaciones nativas iOS y Android. Es una herramienta que quiere ofrecer, en su versión Android, algo que no existe en el mercado en cuanto a accesibilidad móvil. Ha sido lanzada al mercado el 6 de marzo de 2015, durante la fase de desarrollo de este proyecto. En este proyecto se realizará una comparativa de ambos software como parte de las pruebas del mismo.

## 4. Evaluación y estudio de los estándares y guías de accesibilidad web actuales

Las guías de accesibilidad tienen como objetivo vencer las limitaciones visuales, motrices, auditivas y cognitivas, que puedan padecer los usuarios, tales como baja visión, ceguera, daltonismo, dificultad de usar las manos, temblores, distrofia muscular, deficiencias auditivas o sordera, dificultades de aprendizaje...incluso problemas de memoria y atención, que suelen ser muy comunes sobre todo en personas mayores.

La persona con discapacidad suele tener problemas con el manejo de los dispositivos, como puede ser un cajero automático con la importancia que ello conlleva; la interacción con las interfaces, como algo tan aparentemente sencillo como recibir una llamada, puede ser algo difícil para una persona con discapacidades motrices; y el acceso a los contenidos, ya que hoy en día un menú de navegación puede tener una gran cantidad de opciones y no ser demasiado accesible para la mayoría de las personas.

Una página web o una aplicación móvil debe tener contenidos claros y simples, mecanismos sencillos de navegación y con información accesible para todos con independencia de sus limitaciones. Los productos de apoyo o ayudas técnicas con dispositivos empleados por las personas con discapacidad para atenuar, prevenir o neutralizar esa discapacidad. Hay diversas tecnologías de apoyo [**Productos de Apoyo, Glosario de Términos**] que pueden usar los usuarios discapacitados para usar un dispositivo móvil o navegar por una página web, tales como un dispositivo hardware que convierte texto en caracteres Braille (llamado "Líneas Braille"); un lector de pantalla que disponga de síntesis de voz que convierta todo el texto a voz que se observe en la pantalla del dispositivo, incluso los diferentes menús que pueda tener la aplicación; un magnificador de pantalla que amplíe lo que se muestra en la pantalla del dispositivo; y hasta algún programa software de ayuda, que facilite el manejo de un menú en un dispositivo. Todos estos productos de apoyo pueden usarse tanto en entorno web con un ordenador de sobremesa, como con distintos dispositivos táctiles.

El consorcio World Wide Web [W3C, 1994], es el máximo organismo en Internet que se encarga de promover la accesibilidad, en concreto su grupo de trabajo Web Accessibility Initiative [**WAI, Glosario de Términos**], creadores de las pautas de accesibilidad WCAG 1.0



[WCAG 1.0, 1999] y WCAG 2.0 (Pautas de Accesibilidad de Contenido Web 2.0) [WCAG 2.0, 2008]. Estas pautas de accesibilidad se dividen en:

- 1) **Pautas de Accesibilidad al Contenido de la Web (WCAG):** Enfocadas para desarrolladores web y de aplicaciones móviles, e informan de como conseguir que las webs o las apps sean accesibles.
- 2) **Pautas de Accesibilidad para Herramientas de Autor (ATAG):** Enfocadas para evaluadores de software, de forma que con estas pautas se facilite la creación de páginas web y aplicaciones móviles accesibles.
- 3) **Pautas de Accesibilidad para Agentes de Usuario (UAAG):** Enfocadas a los usuarios, tales como navegadores y programas similares, de forma que se mejore la accesibilidad a la web y dispositivos móviles. Entre esas pautas de accesibilidad encontramos las recomendadas por la Web Accessibility Initiative [WAI, "<http://www.w3.org/WAI/guid-tech.html>"].

Como vimos brevemente en el apartado **3.3 Trabajos previos, alternativas posibles y trabajos de otros investigadores**, hay diversas empresas u organizaciones que disponen de sus propias recomendaciones o guías de evaluación de accesibilidad, generalmente enfocadas a una página web, aunque alguna de estas organizaciones dispone o ha adaptado las pautas de accesibilidad de sus guías al entorno móvil.

Las WCAG 2.0 son actualmente un estándar internacional y sus pautas de accesibilidad abarcan un extenso rango de recomendaciones para la creación de contenido web más accesible. Permite a un mayor número de personas con discapacidad (sordera y deficiencias auditivas, ceguera y baja visión, limitaciones cognitivas y de movilidad, , deficiencias del aprendizaje, deficiencias del habla y fotosensibilidad) que, mediante el seguimiento de estas pautas de accesibilidad tengan una mejor experiencia de usuario al interactuar con el contenido web, haciendo que sea más accesible y usable para cualquier persona.

El consorcio W3C [W3C, 1994] dispone de una guía que especifica las Mejores Prácticas para mejorar la experiencia de usuario al acceder a dispositivos móviles [MWBP, "<http://www.w3.org/TR/mobile-bp/>"], y también una ayuda al desarrollo de aplicaciones web móviles para que el contenido sea más productivo y dinámico, teniendo en cuenta las pautas de accesibilidad más importantes, centrándose en las que permiten una mejor experiencia de usuario y mostrando la advertencia de las pautas de accesibilidad que se consideran mal aplicadas, [MWABP, "<http://www.w3.org/TR/mwabp/>"]. De estas prácticas y guías de accesibilidad se hablará con mayor detenimiento en unos párrafos más adelante.

El W3C anunció el lanzamiento de la Iniciativa de Web Móvil MWI [MWI, 2005], con fecha de 11 de mayo de 2005, con el objetivo de convertir el acceso a la web desde un dispositivo móvil en una experiencia de usuario similar a un dispositivo de sobremesa. Los participantes de esta iniciativa se centrarán en fomentar buenas prácticas y en mejorar la descripción de los dispositivos móviles, mediante la creación y verificación de pautas de accesibilidad, para conseguir desarrollar mejores soluciones, incorporando una base de datos de descripciones que pueden ser utilizadas por los creadores de contenido para que puedan adaptarse a un dispositivo en concreto. Hay varias empresas comerciales que también están involucradas apoyando esta iniciativa.

Funka Nu [Funka Nu, 2000] se inició como un proyecto por las organizaciones de discapacidad de Suecia. Actualmente dispone de oficinas en Estocolmo, Oslo y Madrid y clientes por todo el mundo. y son cofundadores de la Asociación Internacional de Profesionales de Accesibilidad IAAP. Como vimos en el apartado **3.3 Trabajos previos, alternativas posibles y trabajos de otros investigadores**, fue desarrollando una metodología colaborando con una asociación de personas con discapacidad, y centrándose en pautas de accesibilidad para el entorno móvil. Las guías desarrolladas por Funka Nu [Funka Nu Guidelines, "<http://www.funka.com/contentassets/5f2704e1efa942a29f9bbb20facea4d1/mobile-navigation-guidelines-funka-2014.pdf>"] están basadas en las pautas internacionales del estándar de accesibilidad a los contenidos web (WCAG 2.0). Se ha demostrado que las pautas WCAG 2.0 [WCAG 2.0, "<http://www.sidar.org/traducciones/wcag20/es/>"] no son suficientes para el entorno móvil, de ahí el inicio de este proyecto para Funka Nu con el objetivo de mejorar la accesibilidad en dispositivos móviles táctiles.

BBC [BBC, 2005] es una organización que incorpora una serie de estándares promovidos por la W3C que facilitan el acceso a los contenidos publicados en páginas web a los usuarios que utilizan dispositivos especiales como lectores de pantalla. Dispone de una guía de buenas prácticas de accesibilidad [BBC Guidelines, "<http://www.bbc.co.uk/guidelines/futuremedia/accessibility/mobile>"] para internet móvil. Estas prácticas tienen en cuenta contenido web, aplicaciones híbridas y nativas para dispositivos móviles y está orientado a la experiencia de usuario.

Paul J. Adam es un consultor de accesibilidad en el mundo empresarial, que ha desarrollado una serie de pautas de accesibilidad móvil y web [Paul J. Adam, "<http://pauljadam.com/favelets/>"], centrándose en servicios, presentaciones, demos y aplicaciones. Se centra en como aportar accesibilidad para los distintos elementos que

componen una página web, tales como imágenes, formularios, tablas, enlaces, encabezados y contenido dinámico.

A continuación se describirá, por orden de importancia para este proyecto, cada una de las guías de accesibilidad de las que dispone cada una de las guías de accesibilidad de las que dispone cada organización, y que han sido utilizadas para realizar posteriormente una propuesta de guía de accesibilidad para dispositivos móviles.

Las pautas de accesibilidad descritas por Funka Nu [Funka Nu Guidelines, "<http://www.funka.com/contentassets/5f2704e1efa942a29f9bbb20facea4d1/mobile-navigation-guidelines-funka-2014.pdf>"] son la base de la propuesta de guía de accesibilidad móvil de este proyecto, en las que clasifica dichas pautas en diferentes categorías, en función del tipo de solución elegida, diseño, estructura y presentación, interacción, contenido y configuración de usuario. Las pautas de accesibilidad abarcan desde conseguir que un sitio web funcione en un dispositivo móvil, identificar elementos gráficos, iconos y botones con su función, cada objeto de formulario debe disponer de una etiqueta o descripción, limitar la cantidad de información y número de objetos mostrados, utilizar contrastes altos, control mediante gestos en la interfaz de usuario, proporcionar feedback al usuario, evitar abreviaturas, posibilidad de cambiar el tipo de letra y de hacer zoom a la pantalla del dispositivo para ampliar la imagen....todas estas pautas de accesibilidad son muy importantes y útiles, y sólo están recogidas por Funka Nu de forma clara, ya que otras guías o metodologías no especifica esta funcionalidad accesible de forma tan clara.

No necesariamente serán las mismas categorías de pautas de accesibilidad en que las organiza Funka Nu las que se establecerán más adelante en la propuesta de guía de accesibilidad, ya que se quiere ampliar todo lo posible esta guía teniendo en cuenta otras recomendaciones de distintas organizaciones.

Las pautas de accesibilidad WCAG 2.0 [WCAG 2.0, "<http://www.sidar.org/traduccion/wcag20/es/>"] que propone el consorcio World Wide Web [W3C, 1994] están principalmente orientadas a páginas web, aunque según se realice su interpretación, ciertas pautas de accesibilidad pueden adaptarse al entorno móvil, aunque por lo general, muchas de ellas son demasiado ambiguas y no están lo suficientemente desglosadas para un caso específico, sino que lo trata de forma general. Las pautas de accesibilidad se clasifican en cuatro grupos: perceptible, operable, comprensible y robusto. Estas pautas abarcan desde proporcionar alternativas textuales para todo contenido no textual, acceso a la funcionalidad mediante teclado, hacer los contenidos textuales legibles y



comprensibles e incluir ayudas técnicas para la compatibilidad de aplicaciones, por citar una pauta de accesibilidad de cada categoría por ejemplo.

Las pautas de accesibilidad de la organización BBC [BBC Guidelines, "<http://www.bbc.co.uk/guidelines/futuremedia/accessibility/mobile>"] están orientadas a la experiencia de usuario y enfocadas a los dispositivos móviles. Divide las pautas de accesibilidad en distintas categorías: audio y vídeo, diseño, foco, formularios, imágenes, enlaces, scripts y contenido dinámico y estructura. Para cada pauta asigna si es obligatoria o recomendada, para distinguir distintos tipos de accesibilidad.

Como ya mencionamos anteriormente, el consorcio W3C también dispone de unas prácticas o recomendaciones de accesibilidad, una de ellas es Mobile Web Best Practices 1.0 [MWBP, "<http://www.w3.org/TR/mobile-bp/>"], la cual ayuda al desarrollo de aplicaciones web móviles para que el contenido sea más dinámico y para mejorar la experiencia del usuario en dispositivos móviles. Está muy orientada al desarrollador con recomendaciones para los distintos elementos que componen una página web como navegación y enlaces, estructura de la página y entrada de datos, por ejemplo. No está realmente adaptada para dispositivos móviles, aunque algunas de sus recomendaciones si se pueden aplicar, como veremos en la propuesta de guía de evaluación de accesibilidad.

Igualmente el W3C tiene otras recomendaciones [MWABP, "<http://www.w3.org/TR/mwabp/>"], más centradas en la experiencia de usuario que en el desarrollo de las mismas, en las que se tiene en cuenta los datos de las aplicaciones, la seguridad y privacidad y la redirección de página dentro de una sitio web, por citar algunos ejemplos.

Con esta información se realizará un profundo estudio de todas las pautas, guías y recomendaciones de accesibilidad descritas por estas organizaciones, y se creará una propuesta de guía de evaluación de accesibilidad lo más útil posible para este proyecto y que sirva para otros proyectos igualmente, siempre que estén orientados a accesibilidad. Se determinará qué pautas de accesibilidad de cada guía se pueden aplicar al entorno móvil y cuáles no, así como las que no serán contempladas en este proyecto. posteriormente se mostrará en forma de tabla para que quede mejor reflejado y sea más conciso y comprensible el resultado obtenido. El siguiente apartado contendrá todas las pautas de accesibilidad que deberían aplicarse a una aplicación móvil, y realizando una comparativa con las recomendaciones ofrecidas por las distintas organizaciones en función de su importancia.

## 4.1 Propuesta de guía de evaluación de accesibilidad

De las categorías de las pautas de accesibilidad que se mostrarán en la propuesta de guía de accesibilidad se comentarán brevemente dos de ellas por su gran importancia hoy en día: responsive design (diseño adaptable) y scripts, contenido dinámico y animaciones (WAI-ARIA).

El Responsive Design es un estilo de diseño y desarrollo que tiene como objetivo adaptar la apariencia de una página web a cualquier dispositivo que se esté utilizando para poder visualizarla, con independencia del tamaño de la pantalla del dispositivo. El uso de Media Queries, una serie de órdenes que se incluyen en la hoja de estilos CSS e indican al documento HTML cómo tiene que comportarse en función de las resoluciones de pantalla en las propiedades de las hojas de estilo CSS.

WAI-ARIA 1.0 [WAI-ARIA, 2014] (Web Accessibility Initiative-Accessible Rich Internet Applications) es una especificación del W3C [W3C, 1994], que actualmente está trabajando en WAI-ARIA 1.1. Está pensado para hacer más accesible el contenido dinámico y los controles relacionados con Ajax, HTML, Javascript y tecnologías relacionadas. Permite a las APIs (Interfaz de Programación de Aplicaciones) de accesibilidad recibir información sobre el comportamiento de la interfaz y su estructura, de forma que esta información pueda ser utilizada por los productos de apoyo o ayudas técnicas para la interacción con el usuario final. WAI-ARIA ofrece una serie de roles, estados y propiedades que definen los elementos de la interfaz, permite incluir información semántica sobre la estructura de la página web, para que los productos de apoyo puedan acceder y saltar por los bloques más relevantes de la página o aplicación móvil.

WAI-ARIA permite definir que rol o función tiene un elemento y así poder indicar su estado y propiedades, que se podrán modificar dinámicamente para que los productos de apoyo anuncien los cambios que se van a producir. Por ejemplo, para un menú desplegable se puede indicar si está o no plegado, y cambiar ese estado cuando el usuario interactúe con él para que los productos de apoyo anuncien el nuevo estado del control.

La pautas de accesibilidad propuestas se clasificarán en categorías(1, 2, ...) y subcategorías (1.1, 1.2..), acompañadas de una breve descripción después de nombrar cada una de las mismas.

### 1. GENERAL

#### 1.1 DISEÑO GENERAL



- **1.1.1 Las imágenes, los botones y los iconos deben tener una función específica y una descripción.**
- Las imágenes significativas deben tener un texto alternativo. Las imágenes y los botones de la aplicación deben incluir una descripción.
- **1.1.2 Diseño de botones, enlaces, menús y cabeceras de forma homogénea y específico de dispositivos móviles.**
- Colocar los botones que tengan una determinada funcionalidad en el mismo lugar de la pantalla y diseñarlos de forma homogénea.
- **1.1.3 Agrupación de elementos que van juntos. La página debe redistribuirse para que la información relacionada se posicione justo detrás de la sección con la que se relaciona.**
- No colocar todos los elementos abajo del todo, sino de manera uniforme de la aplicación.
- **1.1.4 Diseño limpio que minimice el número de objetos en pantalla para cargar la aplicación lo antes posible.**
- Los sitios web que han sido diseñados para pantallas grandes disponen de muchos objetos y áreas que, al visualizar esos elementos en una pantalla pequeña, obliga a desplazarse mucho por la pantalla, además del aumento del tiempo de carga en el dispositivo.

## **1.2 RESPONSIVE DESIGN (DISEÑO ADAPTABLE)**

- **1.2.1 Debe visualizarse la interfaz tanto en horizontal como en vertical.**
- **1.2.2 Mostrar un menú sencillo y en diferentes formas dependiendo del ancho de la pantalla. Los menús desplegados deben tener una funcionalidad clara e intuitiva que minimice la información y texto mostrados.**
- En los dispositivos móviles los menús deben diseñarse de modo que ocupen poco espacio y tengan una disposición visual clara. En algunas situaciones puede ser necesario mostrar el menú de diferentes formas, en función del ancho de pantalla.

- ej: "dropdown menu", una lista desplegable en la que el desplazamiento se realiza con un selector.
- **1.2.3 Longitud de línea adaptada al ancho de pantalla, entre 55 y 70 caracteres, incluido espacios.**
- **1.2.4 Implementar una alternativa para la página cuando no está soportado o permitido ningún lenguaje de script. La combinación de CSS y marcadores descriptivos hace que la página se vea correctamente con o sin hojas de estilo.**

### 1.3 ESTRUCTURA

- **1.3.1 Orientación de botones y enlaces en filas.**
- Los usuarios que no ven la interfaz pueden localizar los botones más fácilmente, de forma que si el usuario encuentra un botón, es más fácil que encuentre los demás, estén orientados en horizontal o en vertical. Además crea una impresión visual más clara para usuarios videntes.
- **1.3.2 Objetos para clicar fáciles de usar, con espacios suficientes de separación entre ellos. Reutilizar colocaciones conocidas de los iconos.**
- ej: si la pantalla está expuesta a la luz solar directa, y únicamente se ha utilizado el color para indicar que hay enlaces, será difícil visualizar esos enlaces.
- **1.3.3 Uso de encabezados breves y descriptivos para estructurar la información e indicar la estructura lógica del documento y cierre de cada una de las etiquetas abiertas en lenguajes para dispositivos móviles y en lenguajes de marcado.**
- ej: uso de etiquetas <h1>...<h6> en HTML, para lenguaje de marcado.
- ej: cierre de sintaxis en otros lenguajes como Java o Go.
- **1.3.4 Título de página breve y descriptivo.**
- **1.3.5 Evitar el desplazamiento horizontal.**
- **1.3.6 Identificar el idioma utilizado en todo momento.**
- No sólo identificar el idioma general de la página, sino también indicarle al lector de pantalla si hay texto o si se escribe en otro idioma.

## 1.4 PRESENTACIÓN

- **1.4.1 Separar la presentación del contenido.**
  - ej: con las hojas de estilo CSS se puede crear un fichero que incluya distintos formatos de presentación que se pueda cambiar con independencia del contenido en HTML u otro lenguaje de marcado.
  - ej: patrón de diseño MVC (modelo-vista-controlador) en Java, siendo la vista la presentación, el modelo el contenido y el controlador para la gestión de eventos y acceso a datos.
- **1.4.2 Tener hojas de estilo que no ocupen demasiado espacio.**
  - ej: hojas de estilo CSS.

## 2. COLOR

- **2.1 Permitir configurar colores. Elección de colores de fondo y primer plano con contrastes adecuados.**
  - ej: background y foreground deben tener suficiente contraste para que el contenido siga siendo legible.
- **2.2 Utilizar contrastes altos.**
  - Facilita ver correctamente la pantalla a plena luz solar.
- **2.3 Asegurarse de que la información transmitida a través del color está disponible también sin color.**
  - Las personas que sufren de dificultad para distinguir los colores o daltonismo, les será muy útil la posibilidad de ver la información en blanco y negro.

## 3. TEXTO

- **3.1 Alternativas textuales. Ofrecer un texto equivalente para todo elemento no textual.**
  - El contenido multimedia como vídeos, audio, imágenes debe de disponer de una descripción en modo texto, con la excepción de algunos casos como los captchas.

- **3.2 Permitir configurar el tipo y tamaño de letra. Función de ampliación (zoom) de la interfaz para texto.**
- **3.3 Uso correcto de tipografía y tamaño de letra para que sea legible.**
- Todo texto puede ser ajustado sin ayudas técnicas hasta un 200% sin perder el contenido o la funcionalidad, a excepción de los subtítulos y las imágenes de texto.

#### **4. CONTENIDO ESTÁTICO**

##### **4.1 IMÁGENES**

- **4.1.1 Minimizar el número de imágenes y colocarlas en el código con un enlace que haga referencia al directorio dónde se encuentran esas imágenes. Incluir imágenes decorativas de fondo embebidas en las hojas de estilo. Establecer la anchura y altura de las imágenes en las hojas de estilo independientemente de su anchura y altura real.**
- ej: imágenes decorativas en hojas de estilo CSS, imágenes en código HTML pero se modifica su anchura y altura en las hojas de estilo CSS.
- **4.1.2 Función de ampliación (zoom) de la interfaz para imagen.**

##### **4.2 TABLAS**

- **4.2.1 No usar tablas anidadas.**
- **4.2.2 Utilizar una alternativa a las tablas cuando sea posible.**
- Las tablas no funcionan bien en tamaños limitados como el de una pantalla de un dispositivo móvil, ya que obliga al usuario a desplazarse horizontalmente para leer los contenidos de la tabla.
- **4.2.3 Incluir un título que proporcione una descripción breve de la tabla.**
- **4.2.4 Incluir un resumen con una descripción larga con explicación sobre el contenido y la estructura de la tabla.**
- Explicar el objetivo de la tabla, incluir el número de filas y columnas que contiene y una descripción de los encabezamientos.
- **4.2.5 Definir los encabezamientos de las columnas y las filas.**

- **4.2.6 Utilizar etiquetas y atributos especiales para asociar las celdas de encabezamiento y las celdas de datos para los encabezamientos más complejos.**
- ej: uso de los atributos "scope" y "headers" para definir la relación que existe entre las celdas de encabezamiento y las celdas de datos.

### 4.3 FORMULARIOS

- **4.3.1 a) Campos de formulario con etiqueta visible asociada o descripción en texto del campo. Las etiquetas de los campos de introducción deben colocarse encima del campo.**
- Los objetos de formulario como campos de texto, deben tener esa etiqueta visible, generalmente indicado con el elemento "label".
- **4.3.1 b) Los botones de opción y las casillas de verificación pueden tener texto situado a la derecha y también pueden tener un título encima de ellos que indique su función.**
- ej: "tooltip", "hint attribute".
- **4.3.2 Recordar los últimos datos introducidos por el usuario, si es posible.**
- **4.3.3 Crear un orden lógico a través de enlaces, controles de formulario y objetos.**
- Estos componentes deben preservar su significado si se navega de forma secuencial por la página web o aplicación móvil.
- **4.3.4 Permitir configurar un tiempo predeterminado al usuario para el relleno de formularios.**
- La aplicación debe proporcionar al usuario el tiempo suficiente para introducir textos, y avisarle cuando se vaya a agotar ese tiempo, incluso dar la opción de prorrogar ese tiempo.
- ej: cierres de sesión automáticos cuando transcurren unos minutos de inactividad en la aplicación.

### 4.4 LOGIN Y CONTRASEÑA

- **4.4.1 Permitir inicio de sesión automático.**



- ej: web2access.
- **4.4.2 Ayuda contextual para la introducción de contraseñas.**
- Asociar semánticamente el campo "contraseña" con el texto informativo sobre sus requisitos, para asegurarse de que será leído por el usuario evitando que cometa un error al rellenar el campo.
- **4.4.3 Incluir una casilla de forma que al marcarla o desmarcarla, muestre en claro la contraseña al escribirla o muestre asteriscos por cada carácter introducido de la misma.**
- **4.4.4 Permitir configurar un tiempo predeterminado al usuario para cerrar sesión.**

## **5. NAVEGACIÓN Y ENLACES**

- **5.1 Insertar accesos directos para saltar de una parte del contenido a otra en páginas largas.**
- Esos accesos directos deben estar ocultos en el estado inicial, mediante bloques "bypass".
- **5.2 Asignar teclas de acceso rápido a enlaces en los menús de navegación y funcionalidad de acceso frecuente.**
- **5.3 Minimizar el número de enlaces en la página.**

## **6. PANTALLA TÁCTIL**

- **6.1 Permitir controlar la interfaz con un solo dedo.**
- ej: ocultar/reaparecer botones al tocar una zona de la pantalla.
- **6.2 Menús e índices visibles al cargar la página.**
- **6.3 Encabezado de página pequeño.**
- **6.4 Áreas grandes para clicar .**
- El tamaño de pulsación sobre la pantalla debe tener como referencia el mínimo del alto de la fila del texto por el alto de la fila del texto multiplicado por 3. Para los iconos



de la aplicación, se puede coger de referencia un mínimo de 9 milímetros de ancho y de alto.

- **6.5 Los botones de control físicos deben de poder usarse para navegar por la interfaz de la aplicación para un sistema operativo o dispositivo móvil.**
- ej: botón "Atrás".
- **6.6 Minimizar la introducción de texto en la interfaz. Ofrecer valores preseleccionados cuando sea posible. Autocompletado y sugerencias de búsqueda.**
- La introducción de texto en dispositivos móviles es compleja y requiere bastante tiempo, por lo que si se usan listas con opciones en lugar de escribir el texto y se usa una función de autocompletado o predicción, se facilita la interacción al usuario y ayuda a evitar errores.
- **6.7 Implementar un control mediante gestos si la interfaz lo permite.**
- ej: deslizar el dedo al pasar páginas, acercar/alejar zoom.
- **6.8 Controlar la interfaz del dispositivo mediante teclado si está preparado para ello.**
- **6.9 No incluir funciones que sólo se puedan ejecutar mediante gestos. Incluir un botón o enlace de complemento.**
- **6.10 Evitar pop-ups u otras ventanas emergentes.**
- **6.11 Informar al usuario del cambio de contexto.**
- ej: abrir una nueva ventana o pop-up, incluir un botón para iniciar el cambio de contexto en lugar de realizarlo de forma automática...
- **6.12 La versión web de la aplicación y la versión móvil incluyen enlaces que permiten cambiar de una versión a otra.**
- **6.13 Adaptar el teclado virtual al contenido que se va a introducir.**
- ej: seleccionar ítems utilizando teclas de navegación y entrada numérica, usar HTML5 para marcar distintos tipos de campos de introducción...
- **6.14 Minimizar el número de pulsaciones de teclas.**

**7. NOTIFICACIONES**

- **7.1 Proporcionar feedback al usuario mediante un led de color, sonido o vibración, con posibilidad de desconectar esta respuesta.**
- Si el usuario realiza una acción, ya sea introducción de texto en la interfaz, un comando de voz o un gesto con el dispositivo, debe obtener una respuesta visual, sonora o háptica (vibración al tocar la pantalla táctil) a dicha acción, sin que existan tampoco demasiadas respuestas de este tipo para que no moleste en exceso ni perjudique la experiencia de usuario.
- **7.2 Ofrecer información de estado clara al usuario durante la interacción.**
- ej: información visual/auditiva en un proceso de carga.
- ej: mostrar un mensaje de error ante una anomalía en la funcionalidad de la aplicación.
- **7.3 Distinguir distintos tipos de notificaciones.**
- ej: avisos, errores, alertas...

**8. SCRIPTS, CONTENIDO DINÁMICO Y ANIMACIONES (WAI-ARIA)**

- **8.1 Incluir información semántica sobre la estructura de la página, los elementos de la misma, su rol y su función. Conservar la estructura semántica.**
- Esto contribuye a que los productos de apoyo puedan anunciar y acceder a bloques importantes de la web o la app móvil, como la cabecera, menú de navegación, el contenido principal, pie de página, el buscador...
- ej: dos tipos de roles, los que definen elementos de la interfaz (árboles, alertas...) y los que definen la estructura de la página (cabecera, navegación, pie).
- ej: dentro de los roles de estructura de la página -> landmark roles -> se usan para identificar áreas separadas de la página.
- **8.2 Informar sobre la actualización del contenido y el contexto de la página sin la intervención del usuario. Modificar dinámicamente los elementos para que los productos de apoyo anuncien los cambios que se produzcan.**

- Las páginas web o apps móviles pueden tener zonas que se actualizan de forma automática, incluyendo o eliminando contenido, incluso modificando y actualizando sin la intervención del usuario. Estas zonas suelen incluir los últimos mensajes de un chat, mensajes de validación de un formulario según el usuario va rellenando sus campos, un banner publicitario que muestra un mensaje que cambia cada ciertos minutos...
- Cuando se realiza la actualización de este contenido, el usuario puede tener el foco en cualquier parte de la página web o app móvil, de forma que si se accede con un producto de apoyo, la actualización será transparente al usuario y le pasará de forma desapercibida.
- ej: uso de una "live-region" para anunciar un cambio automático en la página.
- ej: aria-live, aria-label.
- ej: actualización de contenido, "refresh"; actualización de contexto, abrir nueva página.
- **8.3 Verificar que el rol de un elemento dinámico es fijo y no se cambia, y los estados y propiedades son dinámicos. Cambiar los estados y propiedades en respuesta a los eventos.**
- **8.4 Usar marcado nativo cuando sea posible.**
- WAI-ARIA sirve para mejorar, no para sustituir el marcado nativo. Procurar usar con preferencia un rol nativo que un rol ARIA.
- ej: usar `<input type="checkbox">` o `<button>` en lugar de `<div role="checkbox">`.
- **8.5 Buscar relaciones entre los elementos y marcarlas con el atributo más apropiado.**
- Construir relaciones sólo donde sea necesario, y tener en cuenta que hay algunas relaciones que son automáticas, como la de un "label" con su "input".
- ej: uso de "aria-controls" para las pestañas.
- **8.6 Sincronizar la interfaz visual con la interfaz accesible.**
- Un menú puede presentarse de diferentes formas sin perder información o estructura.

- ej: css attribute selectors.

## **9. MULTIMEDIA (AUDIO Y VÍDEO)**

- **9.1 Alternativa a audio y vídeo grabado, con subtítulos para audio grabado y autodescripción del contenido multimedia.**
- **9.2 Proporcionar subtítulos para todo el contenido de audio en directo.**
- **9.3 Ofrecer un mecanismo para pausar o detener el audio o vídeo.**
- ej: cuando suena el audio de una página web automáticamente más de 3 segundos.

## **10. DOCUMENTACIÓN Y VALIDACIÓN**

- **10.1 Crear documentos que estén validados por las gramáticas formales publicadas.**
- ej: documento que describa técnicas para la creación de contenidos accesibles basado en las guías de accesibilidad.
- **10.2 Verificar que existe documentación de ayuda técnica.**
- El usuario debe disponer de documentación para los productos de apoyo o ayudas técnicas, para que pueda usar con facilidad un lector de pantalla con síntesis de voz, un magnificador de pantalla que aumenta el tamaño del contenido..., etc.
- **10.3 Ofrecer algún modo de ayuda contextual.**
- Al usuario le es muy útil ciertos atajos de teclado o teclas de acceso directo a algún tipo de ayuda técnica o información de la aplicación que está usando en ese momento.
- ej: tecla F1, al pulsarla se abre una ventana de ayuda para consultar dudas del sistema.
- **10.4 La documentación y la ayuda debe estar redactada de la forma más clara y sencilla posible.**
- **10.5 La documentación debe estar disponible en formatos alternativos bajo petición del usuario, ajustándose a sus necesidades específicas.**
- ej: formato pdf, .doc..., etc.

A continuación se muestra en la Tabla 13, las categorías, subcategorías y pautas de la propuesta de guía de evaluación de accesibilidad de forma más esquemática, y realizando una

comparativa entre las diferentes guías y recomendaciones de accesibilidad en las que se basa, de forma que quede más claro el origen de cada pauta de accesibilidad, qué guías tienen en común la misma pauta y cuáles no, y qué pauta está más enfocada a web o a móvil.

Se han tomado como base las guías de accesibilidad de Funka Nu, aunque ampliándolas para conseguir todas las características de accesibilidad a tener en cuenta.

Los términos usados en la Tabla 13 son:

N/A: No se aplica a móvil

Y: Sí se aplica a móvil

N: No se contempla





[illegible]

			para cargar la aplicación lo antes posible												
<b>1. GENERAL</b>	<b>1.2 RESPONSIVE DESIGN (DISEÑO ADAPTABLE)</b>	<b>1.2.1</b>	Debe visualizarse la interfaz tanto en horizontal como en vertical	28)	Y	3.2.3, 3.2.4	Y	--	N	--	N	--	N	--	N
		<b>1.2.2</b>	Mostrar un menú sencillo y en diferentes formas dependiendo del ancho de la pantalla. Los menús desplegables deben tener una funcionalidad clara e intuitiva que minimice la información y texto mostrados	15), 29)	Y	3.2.3, 3.2.4	Y	--	N	--	N	--	N	--	N
		<b>1.2.3</b>	Longitud de línea adaptada al ancho de pantalla, entre 55 y 70 caracteres, incluido espacios	15), 23)	Y	1.4.8	Y	--	N	--	N	--	N	--	N
		<b>1.2.4</b>	Implementar una	--	N	1.3.1	Y	Script y	Y	5.4.9	Y	3.6	Y	--	N

contenido  
dinámico 1)

.4

**1. GENERAL**

**1.3 ESTRUCTURA**

1.3.1

Orientación de botones  
y enlaces en filas

21)

Y

4.1.2

Y

--

N

--

N

--

N

6)

Y

1.3.2

Objetos para clicar  
fáciles de usar, con  
espacios suficientes de  
separación entre ellos.  
Reutilizar colocaciones  
conocidas de los iconos

26)

Y

4.1.2

Y

--

N

--

N

--

N

--

N

1.3.3

Uso de encabezados  
breves y descriptivos  
para estructurar la  
información e indicar la  
estructura lógica del  
documento y cierre de  
cada una de las

17),  
44)

Y

3.2.4, 4.1.1

Y

Estructura 1)

Y

5.4.3

Y

--

N

7)

Y

			etiquetas abiertas en lenguajes para dispositivos móviles y en lenguajes de marcado												
		1.3.4	Título de página breve y descriptivo	--	N	2.4.2	Y	--	N	5.4.1	Y	--	N	--	N
		1.3.5	Evitar el desplazamiento horizontal	--	N	3.2.3	Y	--	N	5.3.3	Y	--	N	--	N
		1.3.6	Identificar el idioma utilizado en todo momento	--	N	3.1.1, 3.1.2	Y	--	N	--	N	--	N	--	N
<b>1. GENERAL</b>	<b>1.4 PRESENTACIÓN</b>	1.4.1	Separar la presentación del contenido	--	N	--	N	--	N	5.4.9	Y	3.4 .7	Y	--	N
		1.4.2	Tener hojas de estilo que no ocupen demasiado espacio	--	N	--	N	--	N	5.4.9	Y	3.4 .7	Y	--	N
<b>2. COLOR</b>		2.1	Permitir configurar colores. Elección de colores de fondo y primer plano con	47)	Y	1.4.8	Y	Diseño 2)	Y	5.3.6	Y	--	N	--	N



			contrastes adecuados												
		2.2	Utilizar contrastes altos	27)	Y	1.4.3, 1.4.6	Y	Diseño 1)	Y	5.3.6	Y	--	N	--	N
		2.3	Asegurarse de que la información transmitida a través del color está disponible también sin color	--	N	1.4.1	Y	--	N	5.3.6	Y	--	N	--	N
<b>3. <u>TEXTO</u></b>		3.1	Alternativas textuales. Ofrecer un texto equivalente para todo elemento no textual	--	N	1.1.1	Y	--	N	5.4.5	Y	--	N	--	N
		3.2	Permitir configurar el tipo y tamaño de letra. Función de ampliación (zoom) de la interfaz para texto	48), 46)	Y	1.4.4	Y	Diseño 6)	Y	--	N	--	N	--	N
		3.3	Uso correcto de tipografía y tamaño de letra para que sea legible	--	N	1.4.4	Y	Diseño 3)	Y	--	N	--	N	--	N
<b>4. <u>CONTENIDO ESTÁTICO</u></b>	<b>4.1 IMÁGENES</b>	4.1.1	Minimizar el número de imágenes y colocarlas en el código con un	43)	Y	--	N	--	N	5.4.6. 5.4.9	Y	--	N	--	N

			enlace que haga referencia al directorio dónde se encuentran esas imágenes. Incluir imágenes decorativas de fondo embebidas en las hojas de estilo. Establecer la anchura y altura de las imágenes en las hojas de estilo independientemente de su anchura y altura real												
		4.1.2	Función de ampliación (zoom) de la interfaz para imagen	46)	Y	1.4.4	Y	Diseño 6)	Y	--	N	--	N	--	N
<b>4. CONTENIDO ESTÁTICO</b>	<b>4.2 TABLAS</b>	4.2.1	No usar tablas anidadas	--	N	--	N	--	N	5.4.4	Y	--	N	5	Y
		4.2.2	Utilizar una alternativa a las tablas cuando sea posible	--	N	--	N	--	N	5.4.4	Y	--	N	5	Y
		4.2.3	Incluir un título que proporcione una descripción breve de la tabla	--	N	--	N	--	N	--	N	--	N	5	Y

		4.2.4	Incluir un resumen con una descripción larga con explicación sobre el contenido y la estructura de la tabla	--	N	--	N	--	N	--	N	--	N	5	Y
		4.2.5	Definir los encabezamientos de las columnas y las filas	--	N	--	N	--	N	--	N	--	N	5	Y
		4.2.6	Utilizar etiquetas y atributos especiales para asociar las celdas de encabezamiento y las celdas de datos para los encabezamientos más complejos	--	N	--	N	--	N	--	N	--	N	5	Y
4. <u>CONTENIDO ESTÁTICO</u>	4.3 FORMULARIOS	4.3.1 a)	Campos de formulario con etiqueta visible asociada o descripción en texto del campo. Las etiquetas de los campos de introducción deben colocarse encima del campo.  Los botones de opción	9), 22)	Y	3.3.3, 3.3.2	Y	Formularios 1), 2)	Y	5.5.3	Y	--	N	4)	Y

		4.3.1 b)	y las casillas de verificación pueden tener texto situado a la derecha y también pueden tener un título encima de ellos que indique su función												
		4.3.2	Recordar los últimos datos introducidos por el usuario, si es posible	--	N	--	N	--	N	5.5.1	Y	--	N	--	N
		4.3.3	Crear un orden lógico a través de enlaces, controles de formulario y objetos	--	N	2.4.3, 2.4.4	Y	Formularios 4)	Y	5.5.2	Y	--	N	--	N
		4.3.4	Permitir configurar un tiempo predeterminado al usuario para el relleno de formularios	41)	Y	2.2.1	Y	--	N	--	N	3.5 .1	Y	--	N
<b>4. CONTENIDO ESTÁTICO</b>	<b>4.4 LOGIN Y CONTRASEÑA</b>	4.4.1	Permitir inicio de sesión automático	--	N	--	N	--	N	--	N	3.3 .2	Y	--	N
		4.4.2	Ayuda contextual para la introducción de contraseñas	--	N	3.3.2. 3.3.5	Y	--	N	--	N	--	N	--	N

		4.4.3	Incluir una casilla de forma que al marcarla o desmarcarla, muestre en claro la contraseña al escribirla o muestre asteriscos por cada carácter introducido de la misma	--	N	3.3.2	Y	--	N	--	N	--	N	--	N
		4.4.4	Permitir configurar un tiempo predeterminado al usuario para cerrar sesión	41)	Y	2.2.1	Y	--	N	--	N	--	N	--	N
<b>5. <u>NAVEGACIÓN Y ENLACES</u></b>		5.1	Insertar accesos directos para saltar de una parte del contenido a otra en páginas largas	32)	Y	2.4.4	Y	Enlaces 1)	Y	5.2.6	Y	--	N	6)	Y
		5.2	Asignar teclas de acceso rápido a enlaces en los menús de navegación y funcionalidad de acceso frecuente	--	N	3.2.3	Y	--	N	5.2.5	Y	--	N	--	N
		5.3	Minimizar el número de enlaces en la página	--	N	--	N	--	N	5.2.9	Y	--	N	--	N



<b>6. PANTALLA TÁCTIL</b>	<b>6.1</b>	<b>Permitir controlar la interfaz con un solo dedo</b>	36)	Y	2.1.1	Y	--	N	--	N	3.5.3	Y	--	N
	<b>6.2</b>	<b>Menús e índices visibles al cargar la página</b>	17)	Y	--	N	--	N	--	N	--	N	7)	Y
	<b>6.3</b>	<b>Encabezado de página pequeño</b>	17)	Y	2.4.6	Y	Estructura 1)	Y	--	N	--	N	7)	Y
	<b>6.4</b>	<b>Áreas grandes para clicar</b>	18)	Y	2.1.1	Y	Diseño 4)	Y	--	N	--	N	--	N
	<b>6.5</b>	<b>Los botones de control físicos deben de poder usarse para navegar por la interfaz de la aplicación para un sistema operativo o dispositivo móvil</b>	30)	Y	--	N	Diseño 7)	Y	--	N	3.5.3	Y	--	N
	<b>6.6</b>	<b>Minimizar la introducción de texto en la interfaz. Ofrecer valores preseleccionados cuando sea posible. Autocompletado y sugerencias de</b>	33), 42)	Y	3.3.5	Y	Notificaciones 1), Recomendaciones 4)	Y	--	N	3.4.2	Y	--	N

			<b>búsqueda</b>												
		<b>6.7</b>	<b>Implementar un control mediante gestos si la interfaz lo permite</b>	34)	Y	2.1.1	Y	--	N	--	N	3.5 .3	Y	--	N
		<b>6.8</b>	<b>Controlar la interfaz del dispositivo mediante teclado si está preparado para ello</b>	31)	N/A	2.1.1, 2.1.2, 2.4.3, 2.4.7	Y	Formularios 3)	Y	--	N	--	N	--	N
		<b>6.9</b>	<b>No incluir funciones que sólo se puedan ejecutar mediante gestos. Incluir un botón o enlace de complemento</b>	35)	Y	2.1.1	Y	--	N	--	N	--	N	--	N
		<b>6.10</b>	<b>Evitar pop-ups u otras ventanas emergentes</b>	--	N	3.2.1, 3.2.2	Y	--	N	5.2.8	Y	3.4 .3	Y	--	N
		<b>6.11</b>	<b>Informar al usuario del cambio de contexto</b>	--	N	3.2.5	Y	--	N	5.2.8	Y	3.4 .3	Y	--	N
		<b>6.12</b>	<b>La versión web de la aplicación y la versión móvil incluyen enlaces que permiten cambiar</b>	1), 2), 3), 4)	Y	--	N	Recomendaciones 1), 2)	Y	--	N	--	N	--	N

			de una versión a otra												
		6.13	Adaptar el teclado virtual al contenido que se va a introducir	11)	Y	2.1.1	Y	Formularios 3)	Y	5.5.1	Y	--	N	--	N
		6.14	Minimizar el número de pulsaciones de teclas	--	N	--	N	--	N	5.5.1	Y	--	N	--	N
<b>7. NOTIFICACIONES</b>		7.1	Proporcionar feedback al usuario mediante un led de color, sonido o vibración, con posibilidad de desconectar esta respuesta	39)	Y	3.3.5	Y	Notificaciones 1)	Y	--	N	--	N	--	N
		7.2	Ofrecer información de estado clara al usuario durante la interacción	40)	Y	3.3.2	Y	Notificaciones 2)	Y	--	N	--	N	--	N
		7.3	Distinguir distintos tipos de notificaciones	--	N	--	N	Notificaciones 3)	Y	--	N	--	N	--	N
<b>8. SCRIPTS, CONTENIDO DINÁMICO Y ANIMACIONES</b>		8.1	Incluir información semántica sobre la estructura de la página, los elementos de la misma, su rol y su	--	N	--	N	--	N	--	N	--	N	8)	Y

<b>(WAI-ARIA)</b>			función. Conservar la estructura semántica												
	8.2		Informar sobre la actualización del contenido y el contexto de la página sin la intervención del usuario. Modificar dinámicamente los elementos para que los productos de apoyo anuncien los cambios que se produzcan	41)	Y	--	N	--	N	5.2.8	Y	3.5 .4	Y	8)	Y
	8.3		Verificar que el rol de un elemento dinámico es fijo y no se cambia, y los estados y propiedades son dinámicos. Cambiar los estados y propiedades en respuesta a los eventos	--	N	--	N	--	N	--	N	--	N	8)	Y
	8.4		Usar marcado nativo cuando sea posible	--	N	--	N	--	N	--	N	--	N	8)	Y

		8.5	Buscar relaciones entre los elementos y marcarlas con el atributo más apropiado	--	N	--	N	--	N	--	N	--	N	8)	Y
		8.6	Sincronizar la interfaz visual con la interfaz accesible	--	N	--	N	--	N	--	N	--	N	8)	Y
<b>9. MULTIMEDIA (AUDIO Y VÍDEO)</b>		9.1	Alternativa a audio y vídeo grabado, con subtítulos para audio grabado y autodescripción del contenido multimedia	--	N	1.2.1, 1.2.2, 1.2.3, 1.2.6	Y	Audio y vídeo 1)	Y	--	N	--	N	--	N
		9.2	Proporcionar subtítulos para todo el contenido de audio en directo	--	N	1.2.4, 1.2.5	Y	Audio y vídeo 1)	Y	--	N	--	N	--	N
		9.3	Ofrecer un mecanismo para pausar o detener el audio o vídeo	--	N	1.4.2	Y	Audio y vídeo 2)	Y	--	N	--	N	--	N
<b>10. DOCUMENTACIÓN Y VALIDACIÓN</b>		10.1	Crear documentos que estén validados por las gramáticas formales publicadas	--	N	--	N	--	N	5.4.7	Y	--	N	--	N



		<b>10.2</b>	<b>Verificar que existe documentación de ayuda técnica</b>	--	N	3.3.5	Y	--	N	--	N	--	N	--	N
		<b>10.3</b>	<b>Ofrecer algún modo de ayuda contextual</b>	42)	Y	3.3.5	Y	--	N	--	N	--	N	--	N
		<b>10.4</b>	<b>La documentación y la ayuda debe estar redactada de la forma más clara y sencilla posible</b>	--	N	3.3.5	Y	--	N	--	N	--	N	--	N
		<b>10.5</b>	<b>La documentación debe estar disponible en formatos alternativos bajo petición del usuario, ajustándose a sus necesidades específicas</b>	--	N	3.3.5	Y	--	N	--	N	--	N	--	N

Tabla 13: Propuesta de guía de evaluación de accesibilidad.



Primero se comentará brevemente algunas carencias observadas en las guías de accesibilidad de Funka Nu, por ser las más importantes ya que son la base de la propuesta de guía de evaluación de accesibilidad, que si se han incorporado a dicha propuesta, para después realizar una síntesis de la misma mostrada en la Tabla 13. Algunas de las pautas de accesibilidad de la propuesta de accesibilidad no están mencionadas en la guías de Funka Nu.

Las carencias más relevantes en las guías de Funka Nu son las que tienen que ver con estas pautas de accesibilidad propuesta en la guía de evaluación: 1.2.4 , 1.3.4 , 1.3.5 , 1.3.6 , 1.4.1, 1.4.2 , 2.3 , 3.1 , 3.3 , 4.2.1 , 4.2.2 , 4.2.3 , 4.2.4 , 4.2.5 , 4.2.6 , 4.3.2 , 4.3.3 , 4.4.1 , 4.4.2 , 4.4.3, 5.2 , 5.3 , 6.10 , 6.11 , 6.14 , 7.3 , 8.1 , 8.3 , 8.4 , 8.5 , 8.6 , 9.1 , 9.2 , 9.3 , 10.1 , 10.2 , 10.4 , 10.5.

Como se puede observar en esas pautas, no hay menciones en algunos aspectos de algunas categorías de pautas de accesibilidad como: responsive design (diseño adaptable), estructura, ninguna mención a las tablas, pocas a la introducción de login y contraseña...y quizás en las categorías que apenas ofrece recomendaciones son las que tienen que ver con WAI-ARIA (contenido dinámico), multimedia (no indica que deba existir alternativas textuales a vídeo, imágenes..) y tampoco hace referencia a ningún tipo de documentación, sea para accesibilidad o para dar soporte a los productos de apoyo.

Las guías de accesibilidad de las demás organizaciones tienen muchas carencias, pero en alguna categoría ofrecen o complementan aspectos de accesibilidad que no están recogidos en las guías de Funka Nu, como pueden ser los aspectos multimedia, bien recogidos en las WCAG 2.0, buenas recomendaciones para notificaciones y formularios ofrecidas por las guías de accesibilidad de la organización BBC, y las recomendaciones de accesibilidad de MWBP, MWABP y Paul J. Adam, complementan alguna pauta de las propuestas en la Tabla 13, de forma que por separado no disponen de una gran colección de pautas de accesibilidad que se puedan aplicar a todos los ámbitos de dispositivos móviles, pero algunas son útiles.

De las pautas no relevantes para accesibilidad móvil descubiertas en Funka Nu, destacan las de minimizar el uso de scripts ejecutados por el cliente y controlar la interfaz de teclado o que la longitud de línea sea de tantos caracteres (entre 55 y 70) haciendo referencia a una pantalla grande, ya que estas pautas no están enfocadas a dispositivos móviles.

Por último, hacer una mención al artículo "WCAG 2.0 Should Not Be Applied to Software and Mobile Apps" [WCAG 2.0 Should Not Be Applied to Software and Mobile Apps, "<https://insight.cryptzone.com/accessibility/wcag-2-0-should-not-be-applied-to-software-and-mobile-apps/>"] que ofrece una visión en la que las pautas de accesibilidad que ofrecen las

WCAG 2.0 no están realmente preparadas para aplicarse tal cual a los dispositivos móviles, al tener diferentes necesidades que un navegador web. Lo más destacable que aporta es que existe una cierta confusión y ambigüedad en estas pautas de accesibilidad para el entorno móvil y se recomienda una API de accesibilidad específica para el dispositivo móvil, algo que no mencionan las WCAG 2.0 y que provocaría que se reduzca el nivel de accesibilidad. Asimismo, asegura que la mejor norma de accesibilidad es "EN 301-549" [UNE-EN 301 549, 2014], con la que se establecen los requisitos funcionales que aseguran que los productos y servicios sean accesibles para todas las personas, se trate de un dispositivo táctil, ordenador de sobremesa o web.

#### 4.2 Recomendaciones de usabilidad

Algunas de las guías de accesibilidad de algunas organizaciones disponen de unas pautas de accesibilidad que están más enfocadas en la usabilidad y la experiencia de usuario y que nos gustaría destacar en este Trabajo Fin de Grado. Se ha recogido estas en forma de recomendaciones, para que una aplicación accesible pueda disponer igualmente de una mejor usabilidad, ya que no impide el acceso, pero sí lo dificulta:

- Minimizar el uso de scripts ejecutados por el cliente, ya que no hay que exigir más de lo necesario al dispositivo.
- La información más importante debe ser colocada más arriba y la menos importante más abajo, de forma que lo más importante de la aplicación o web pueda verse sin desplazarse.
- Los menús e índices deben ser visibles al cargar la página para que el usuario reciba una respuesta rápida a la solicitud de búsqueda web realizada. Asimismo es recomendable un encabezado de página pequeño para no desplazarse mucho por la pantalla.
- Los botones de uso frecuente en el centro y parte de abajo de la pantalla, ya que los usuarios que sólo utilizan una mano o tienen que poner el móvil sobre la rodilla para poder usarlo, les resulta muy complejo pulsar los botones que están en los márgenes.
- Evitar funcionalidades en el extremo derecho de la pantalla, ya que los usuarios con destrezas motoras reducidas tienen dificultades para sujetar el dispositivo móvil.
- Reutilizar diseños de iconos conocidos, para que el usuario intuya con facilidad su funcionalidad.

- Minimizar la introducción de texto en la interfaz. Introducir texto en los dispositivos móviles requiere tiempo y puede tener dificultades, por lo que es mejor si se ofrece el uso de listas con opciones o alguna función de autocompletado o predicción, en la que la interfaz propone ciertas frases cuando detecta que el usuario comienza a introducir texto.
- Usar componentes integrados para su uso específico, como los botones físicos, que deben ser compatibles con la aplicación.
- Evitar abreviaturas, salvo si se explican la primera vez que aparecen.
- Asegurarse de que el tamaño total de la página es apropiada a las limitaciones de memoria del dispositivo, de forma que se forme un cuello de botella [**Cuello de botella, Glosario de Términos**] en el uso de la CPU [**CPU, Glosario de Términos**] o la memoria RAM [**Memoria RAM, Glosario de Términos**] del dispositivo móvil.
- Cambiar las imágenes de tamaño en el servidor para reducir la cantidad de datos transferidos. También es útil crear una nueva regla CSS para cada una de las imágenes de la aplicación.
- Asegurarse de que el contenido se codifica mediante una codificación de caracteres que se sabe que es compatible con el dispositivo, incluso al enviar contenido, debe hacerse en un formato compatible con el dispositivo.
- No habilitar el uso de cookies, por motivos de privacidad y seguridad.
- Eliminar el borde las imágenes con enlace. ya que descuadra las posiciones de los elementos de la aplicación o web.
- No usar colores fuertes para enlaces, ya que los colores llamativos, podría producir irritación visual por los fuertes contrastes de los colores y provocará que los usuarios abandonen la web o aplicación.
- No utilizar lenguaje de marcado para redirigir las páginas automáticamente, sino hacer de uso de algún temporizador.

### 4.3 Conclusiones sobre el estudio de la evaluación de accesibilidad

Se ha hecho un estudio en profundidad de las principales normas y recomendaciones relativas, similares o relacionadas con aplicaciones móviles nativas.



Como ninguna es específica, se ha hecho recomendación de la lista de pautas de accesibilidad mostrada. Ya sabemos que usabilidad y accesibilidad van de la mano, por lo que se ha distinguido algunas pautas de usabilidad que harían que la experiencia de usuario fuera más adecuada si se tienen en cuenta.

Como recomendación, por tanto, para todos los desarrolladores y evaluadores de accesibilidad de aplicaciones móviles, se realiza la propuesta de recomendación, con pautas de accesibilidad a seguir presentada en la página 46.

## 5. Requisitos de accesibilidad generales de la aplicación

Una vez creada la propuesta de guía de evaluación de accesibilidad móvil, a partir de las pautas de accesibilidad descritas en esa propuesta, se interpretarán para adaptarlas para obtener los requisitos de la aplicación. En este apartado, los requisitos de la aplicación están orientados a web móvil, y en el apartado **6. Desarrollo de una herramienta de evaluación de código accesible**, en el subapartado **6.2 Análisis: requisitos para la app móvil nativa**, los requisitos de convertirán de forma que estén orientados a una app móvil nativa.

De los dos tipos de requisitos principales que hay, **requisitos funcionales** y **requisitos no funcionales**, vamos a centrarnos en los funcionales de cara a la futura implementación, que son los que mostrarán la funcionalidad requerida.

Los requisitos no funcionales son una exigencia del sistema o plataforma tecnológica que no se satisface añadiendo código, por lo que no tendrán prioridad en este apartado, aunque se mostrarán tres ejemplos de este tipo de requisito.

Los requisitos funcionales definen la funcionalidad de la aplicación. Se seleccionarán tres categorías de pautas de accesibilidad de la propuesta de guía de evaluación de accesibilidad, que sean realmente importantes para el desarrollo de este proyecto, y se desglosarán los requisitos funcionales derivados de esas pautas.

Las categorías de accesibilidad elegidas son:

6) Pantalla Táctil

7) Notificaciones

8) Scripts, contenido dinámico y animaciones (WAI-ARIA)

Se han elegido estas tres categorías de pautas de la propuesta de guía de evaluación de accesibilidad por su gran importancia en dispositivos móviles, ya que junto a responsive design (diseño adaptable), son las más útiles y en las que más se va avanzar en el futuro desarrollo accesible de apps móviles, en este apartado referente a web móvil. Otras categorías de pautas de accesibilidad también son importantes pero son más generales y se puedan adaptar tanto a una página web como a una aplicación móvil.

Se indicará un ejemplo de requisito no funcional para estas tres categorías de pautas de accesibilidad, y posteriormente se desglosarán los requisitos funcionales de todas las pautas referidas a las mismas.

### REQUISITOS NO FUNCIONALES:

Identificador y Tipo	RSNF-001	No Funcional
Nombre	Minimizar el número de pulsaciones de teclas	

Categoría	Requisito de Pantalla Táctil	
Descripción	La interacción en la pantalla táctil debe realizarse de forma intuitiva y se debe tener acceso al contenido de la página y menús en el mínimo número de pulsaciones que sea posible.	
Necesidad y Prioridad	Obligatorio	Prioridad Media

Identificador y Tipo	RSNF-002	No Funcional
Nombre	No colapsar al dispositivo móvil con muchas notificaciones	
Categoría	Requisito de Notificaciones	
Descripción	Si un dispositivo móvil recibe multitud de notificaciones, se podría comprometer su rendimiento, consumiendo recursos hardware en exceso, ya sea de memoria RAM, memoria caché o batería, llegando a producirse cierta lentitud en dispositivos móviles con pocas prestaciones, lo que afectaría a la usabilidad y accesibilidad del usuario.	
Necesidad y Prioridad	Obligatorio	Prioridad Media

Identificador y Tipo	RSNF-003	No Funcional
Nombre	Evitar información duplicada	
Categoría	Requisito de Wai-Aria	
Descripción	No debe duplicarse información o menús dinámicos para un mismo elemento, así como no se debe repetir una actualización del contenido de la página que no incorpore nueva información.	
Necesidad y Prioridad	Obligatorio	Prioridad Alta

**REQUISITOS FUNCIONALES:**

Identificador y Tipo	RUF-001	Funcional
Nombre	Visualizar menús e índices al cargar la página	
Categoría	Requisito de Pantalla Táctil	

Descripción	Los elementos de la interfaz de usuario que sean menús e índices deben ser visibles nada más cargar la página, dando prioridad a estos elementos antes que a otros como un control de reproducción de vídeo.	
Necesidad y Prioridad	Obligatorio	Prioridad Alta

Identificador y Tipo	RUF-002	Funcional
Nombre	Permitir controlar la interfaz con un solo dedo	
Categoría	Requisito de Pantalla Táctil	
Descripción	Al tocar una zona de la pantalla deben reaparecer u ocultarse los botones que despliegan los menús o cualquier tipo de interacción con la pantalla.	
Necesidad y Prioridad	Obligatorio	Prioridad Alta

Identificador y Tipo	RUF-003	Funcional
Nombre	Autocompletar	
Categoría	Requisito de Pantalla Táctil	
Descripción	La aplicación debe disponer de la función de autocompletar para la edición de texto o para reducir la necesidad de escribir la opción completa en un control de selección, de forma que ofrezca valores preseleccionados escritos con anterioridad.	
Necesidad y Prioridad	Obligatorio	Prioridad Alta

Identificador y Tipo	RUF-004	Funcional
Nombre	Verificar la existencia de un encabezado de página pequeño	
Categoría	Requisito de Pantalla Táctil	
Descripción	Debe comprobarse si el encabezado de página tiene ciertas medidas de ancho y alto en función del tamaño de la pantalla del dispositivo móvil.	
Necesidad y Prioridad	Obligatorio	Prioridad Alta

Identificador y Tipo	RUF-005	Funcional
Nombre	Ofrecer acceso rápido o atajos	
Categoría	Requisito de Pantalla Táctil	
Descripción	Se deben proporcionar combinaciones de teclas para acceder rápidamente a las funciones principales. Estas combinaciones de teclas deben estar documentadas. Ej: abrir fichero con CTRL+A.	
Necesidad y Prioridad	Obligatorio	Prioridad Alta

Identificador y Tipo	RUF-006	Funcional
Nombre	Permitir uso de métodos abreviados de teclado	
Categoría	Requisito de Pantalla Táctil	
Descripción	Las etiquetas de los controles de la interfaz de usuario deben tener mnemotécnicos para un acceso rápido por teclado. Ej: En los menús, el listado de opciones puede mostrar un elemento con una letra subrayada que indica que al presionar la tecla ALT junto a la tecla correspondiente a la letra subrayada, se producirá el mismo efecto que al clicar en ese elemento de menú.	
Necesidad y Prioridad	Obligatorio	Prioridad Baja

Identificador y Tipo	RUF-007	Funcional
Nombre	Definir área táctil de la pantalla	
Categoría	Requisito de Pantalla Táctil	
Descripción	Cada zona táctil de la pantalla debe tener una dimensión óptima de 9 milímetros de alto por 9 de ancho. No debe ser inferior de 8 milímetros de ancho y 7 de alto. La separación entre elementos debe ser como mínimo de 1 milímetro.	
Necesidad y Prioridad	Obligatorio	Prioridad Alta

Identificador y Tipo	RUF-008	Funcional
----------------------	---------	-----------



Nombre	Evitar doble clic	
Categoría	Requisito de Pantalla Táctil	
Descripción	Emular el clic múltiple mediante la pulsación única de una tecla.	
Necesidad y Prioridad	Obligatorio	Prioridad Alta

Identificador y Tipo	RUF-009	Funcional
Nombre	Permitir pulsación prolongada	
Categoría	Requisito de Pantalla Táctil	
Descripción	Emular la pulsación prolongada de un botón de la pantalla mediante la pulsación única de un botón.	
Necesidad y Prioridad	Obligatorio	Prioridad Alta

Identificador y Tipo	RUF-010	Funcional
Nombre	Permitir alternativas para pulsaciones simultáneas de teclas	
Categoría	Requisito de Pantalla Táctil	
Descripción	Debido a que hay personas que no pueden realizar esta acción de forma simultánea, hay que ofrecer alternativas para pulsaciones simultáneas de teclas y botones de la pantalla (puede ser mediante gestos o de otra forma).	
Necesidad y Prioridad	Obligatorio	Prioridad Alta

Identificador y Tipo	RUF-011	Funcional
Nombre	Usar los botones de control físicos para navegar por la interfaz de la aplicación	
Categoría	Requisito de Pantalla Táctil	
Descripción	Los botones de control físicos, como el botón "Atrás", "Escritorio", incluso los botones de volumen (para hacer zoom), deben ser una opción más para navegar por la aplicación.	
Necesidad y Prioridad	Obligatorio	Prioridad Media

Identificador y Tipo	RUF-012	Funcional
Nombre	Implementar un control mediante gestos si la interfaz lo permite	
Categoría	Requisito de Pantalla Táctil	
Descripción	Deslizar el dedo para recibir llamadas, desbloquear el terminal, pasar páginas, acercar o alejar el zoom...	
Necesidad y Prioridad	Obligatorio	Prioridad Alta

Identificador y Tipo	RUF-013	Funcional
Nombre	Incluir enlace o botón de complemento para la funcionalidad	
Categoría	Requisito de Pantalla Táctil	
Descripción	No deben incluirse funciones que solo se puedan ejecutar mediante gestos, tiene que incluirse un botón o enlace de complemento.	
Necesidad y Prioridad	Obligatorio	Prioridad Alta

Identificador y Tipo	RUF-014	Funcional
Nombre	Evitar ventanas emergentes o pop-ups	
Categoría	Requisito de Pantalla Táctil	
Descripción	Es muy útil para el usuario que al pulsar un botón se inicie el cambio de contexto, en vez de realizarlo de forma automática, evitando de esta forma que aparezcan ventanas emergentes de forma inesperada.	
Necesidad y Prioridad	Obligatorio	Prioridad Alta

Identificador y Tipo	RUF-015	Funcional
Nombre	Adaptar el teclado virtual al contenido que se va a introducir	
Categoría	Requisito de Pantalla Táctil	
Descripción	Al pulsar o seleccionar elementos de un menú utilizando las teclas de navegación o la entrada numérica, el uso de HTML5 puede	

	controlar el tipo de datos de entrada en los distintos campos de introducción de texto.	
Necesidad y Prioridad	Obligatorio	Prioridad Alta

Identificador y Tipo	RUF-016	Funcional
Nombre	Proporcionar feedback al usuario mediante un led de color	
Categoría	Requisito de Notificaciones	
Descripción	Cuando el usuario realiza una acción, recibe una respuesta visual si este tipo de respuesta es adecuada a dicha acción (mensaje recibido, llamada perdida, batería baja, poca memoria RAM, falta de espacio en disco...).	
Necesidad y Prioridad	Obligatorio	Prioridad Alta

Identificador y Tipo	RUF-017	Funcional
Nombre	Proporcionar feedback al usuario mediante un sonido	
Categoría	Requisito de Notificaciones	
Descripción	Cuando el usuario realiza una acción, recibe una respuesta sonora si este tipo de respuesta es adecuada a dicha acción (llamada entrante, mensaje recibido, batería baja, inicio de sesión en aplicación...).	
Necesidad y Prioridad	Obligatorio	Prioridad Alta

Identificador y Tipo	RUF-018	Funcional
Nombre	Proporcionar feedback al usuario mediante una vibración	
Categoría	Requisito de Notificaciones	
Descripción	Cuando el usuario realiza una acción, recibe una respuesta vibratoria si este tipo de respuesta es adecuada a dicha acción y si se ha configurado el dispositivo de forma que se permitan este tipo de respuesta (llamada entrante, mensaje recibido, batería baja, pulsación de teclas, actualizaciones...).	

Necesidad y Prioridad	Obligatorio	Prioridad Alta
-----------------------	-------------	----------------

Identificador y Tipo	RUF-019	Funcional
Nombre	Configurar el tipo de respuesta que se ofrecerá como feedback al usuario	
Categoría	Requisito de Notificaciones	
Descripción	El usuario puede activar/desactivar qué tipos de respuesta (visual, sonora, vibración) desea recibir del dispositivo móvil al realizar alguna acción o al recibir notificaciones del mismo.	
Necesidad y Prioridad	Obligatorio	Prioridad Alta

Identificador y Tipo	RUF-020	Funcional
Nombre	Ofrecer información de estado al usuario	
Categoría	Requisito de Notificaciones	
Descripción	Mostrar información del estado del dispositivo al usuario, tal como una respuesta visual o auditiva en un proceso d carga, poco espacio en disco, poca memoria para las aplicaciones...etc.	
Necesidad y Prioridad	Obligatorio	Prioridad Alta

Identificador y Tipo	RUF-021	Funcional
Nombre	Gestionar notificación de aviso	
Categoría	Requisito de Notificaciones	
Descripción	El usuario debe recibir en segundo plano una notificación de aviso sin que se interrumpa la actividad que esté realizando en ese momento.	
Necesidad y Prioridad	Obligatorio	Prioridad Alta

Identificador y Tipo	RUF-022	Funcional
Nombre	Recibir notificación de error	

Categoría	Requisito de Notificaciones	
Descripción	El usuario debe recibir en segundo plano una notificación de error interrumpiendo la actividad que esté realizando en ese momento.	
Necesidad y Prioridad	Obligatorio	Prioridad Alta

Identificador y Tipo	RUF-023	Funcional
Nombre	Recibir notificación de alerta	
Categoría	Requisito de Notificaciones	
Descripción	El usuario debe recibir en segundo plano una notificación de alerta sin que se interrumpa la actividad que esté realizando en ese momento, si la alerta es considera de grado leve. En cambio, si es de carácter urgente o crítico para el dispositivo, se interrumpirá la actividad que esté realizando el usuario para recibir este tipo de notificación.	
Necesidad y Prioridad	Obligatorio	Prioridad Alta

Identificador y Tipo	RUF-024	Funcional
Nombre	Identificar elementos de la interfaz	
Categoría	Requisito de Wai-Aria	
Descripción	Todos los elementos de la interfaz están perfectamente identificados y son únicos en su contexto, con información de su nombre, estado y rol.	
Necesidad y Prioridad	Obligatorio	Prioridad Alta

Identificador y Tipo	RUF-025	Funcional
Nombre	Pausar o detener la presentación de información	
Categoría	Requisito de Wai-Aria	
Descripción	El usuario debe poder pausar o detener la información presentada a través de un carrusel o que se actualice periódicamente en un área de la pantalla (como puede ser pausar o detener la reproducción de	



	un vídeo).	
Necesidad y Prioridad	Obligatorio	Prioridad Alta

Identificador y Tipo	RUF-026	Funcional
Nombre	Controlar el tiempo de presentación de información	
Categoría	Requisito de Wai-Aria	
Descripción	El usuario puede tener la opción de controlar el tiempo de la información que es presentada mediante un carrusel o de las actualizaciones periódicas en un área de la pantalla (como puede ser aumentar el tiempo de sesión sin que se pulse la pantalla del dispositivo).	
Necesidad y Prioridad	Opcional	Prioridad Baja

Identificador y Tipo	RUF-027	Funcional
Nombre	Ofrecer alternativas a imágenes animadas	
Categoría	Requisito de Wai-Aria	
Descripción	Cuando se presentan animaciones debe ofrecerse una versión alternativa no animada de su contenido.	
Necesidad y Prioridad	Obligatorio	Prioridad Alta

Identificador y Tipo	RUF-028	Funcional
Nombre	Usar el atributo TABINDEX para acceder con el tabulador a un elemento de la página	
Categoría	Requisito de Wai-Aria	
Descripción	Si el atributo tabindex="0" permite que un elemento que no coge el foco por defecto pueda recibirlo. Si el atributo tabindex=-1 no se podrá tabular hasta el elemento, aunque mediante Javascript podría cogerse el foco con .focus(). En cambio, en un elemento LI de una lista, se se incluyen los eventos onclick y onkeypress, no se puede acceder al elemento mediante el teclado porque nunca recibirá el foco, como puede ocurrir a los usuarios con lectores de pantalla.	

Necesidad y Prioridad	Obligatorio	Prioridad Alta
-----------------------	-------------	----------------

Identificador y Tipo	RUF-029	Funcional
Nombre	Incluir información sobre el rol o función de un elemento de la página mediante la etiqueta ROLE	
Categoría	Requisito de Wai-Aria	
Descripción	Se incluye a la etiqueta del elemento "role"(por ejemplo: <div role="progressbar">).	
Necesidad y Prioridad	Obligatorio	Prioridad Alta

Identificador y Tipo	RUF-030	Funcional
Nombre	Permitir a un elemento dinámico cambiar de estado	
Categoría	Requisito de Wai-Aria	
Descripción	Los elementos dinámicos como un elemento de un árbol, pueden estar plegados o desplegados, mediante la etiqueta ARIA-EXPANDED.	
Necesidad y Prioridad	Obligatorio	Prioridad Alta

Identificador y Tipo	RUF-031	Funcional
Nombre	Identificar una zona dinámica del contenido que se actualiza automáticamente	
Categoría	Requisito de Wai-Aria	
Descripción	Con las etiquetas ARIA-LIVE, indicaremos cuando queremos que se anuncie la actualización. Con la etiqueta ARIA-ATOMIC se indica si queremos que se anuncie en toda la región o sólo las partes que cambian, y con la etiqueta ARIA-RELEVANT se indica el tipo de actualización que queremos que se anuncie. También se pueden usar otras opciones como ARIA-BUSY cuando muchas partes de un mismo elemento van a ser modificadas.	
Necesidad y Prioridad	Obligatorio	Prioridad Alta

Identificador y Tipo	RUF-032	Funcional
Nombre	Etiquetar un elemento	
Categoría	Requisito de Wai-Aria	
Descripción	Uso de las etiquetas ARIA-LABEL, ARIA-LABELLEDLY y ARIA-DESCRIBEDLY.	
Necesidad y Prioridad	Obligatorio	Prioridad Alta

Identificador y Tipo	RUF-033	Funcional
Nombre	Mejorar la accesibilidad de las sugerencias en los campos de formularios	
Categoría	Requisito de Wai-Aria	
Descripción	Uso de las etiquetas ARIA-AUTOCOMPLETE y ARIA-ACTIVEDESCENDANT.	
Necesidad y Prioridad	Obligatorio	Prioridad Alta

Identificador y Tipo	RUF-034	Funcional
Nombre	Ocultar elementos a los productos de apoyo	
Categoría	Requisito de Wai-Aria	
Descripción	Uso de etiquetas ARIA-HIDDEN o DISPLAY:NONE.	
Necesidad y Prioridad	Opcional	Prioridad Baja

Identificador y Tipo	RUF-035	Funcional
Nombre	Definir un estado nuevo para los checkbox	
Categoría	Requisito de Wai-Aria	
Descripción	Uso de la etiqueta ARIA-CHECKED. Útil para marcar o desmarcar todos los checks de una lista o para indicar que no todos están marcados o desmarcados.	
Necesidad y Prioridad	Obligatorio	Prioridad Alta

Identificador y Tipo	RUF-036	Funcional
Nombre	Verificar que el rol de un elemento dinámico es fijo	
Categoría	Requisito de Wai-Aria	
Descripción	No debe cambiarse de forma dinámica el rol de un elemento. Para forzar a cambiar se tendría que eliminar del DOM del elemento y crear un nuevo elemento con el rol, pero es una práctica no recomendada.	
Necesidad y Prioridad	Obligatorio	Prioridad Alta

Identificador y Tipo	RUF-037	Funcional
Nombre	Usar marcado nativo cuando sea posible	
Categoría	Requisito de Wai-Aria	
Descripción	Uso de <code>&lt;input type="checkbox"&gt;</code> o <code>"button"</code> en lugar de <code>&lt;div role="checkbox"&gt;</code> cuando sea posible.	
Necesidad y Prioridad	Obligatorio	Prioridad Media

Identificador y Tipo	RUF-038	Funcional
Nombre	Marcar las relaciones entre los elementos con el atributo más apropiado	
Categoría	Requisito de Wai-Aria	
Descripción	Uso de las etiquetas ARIA-CONTROLS para las pestañas.	
Necesidad y Prioridad	Obligatorio	Prioridad Alta

Identificador y Tipo	RUF-039	Funcional
Nombre	Sincronizar la interfaz visual con la interfaz accesible	
Categoría	Requisito de Wai-Aria	
Descripción	Se realiza mediante css attribute selectors (ej: <code>[class="top"]</code> ) {	

	background: yellow }.	
Necesidad y Prioridad	Obligatorio	Prioridad Alta

Identificador y Tipo	RUF-040	Funcional
Nombre	Proporcionar mecanismos redundantes de interacción para el manejo de scripts dinámicos	
Categoría	Requisito de Wai-Aria	
Descripción	Uso de etiquetas "onclick" combinadas con etiquetas "onkeypress".	
Necesidad y Prioridad	Opcional	Prioridad Baja



## 6. Desarrollo de una herramienta de valuación de código accesible

### 6.1 Introducción: versión de Android y características de accesibilidad

La versión de Android que se va a utilizar como plataforma tecnológica **[Plataforma Tecnológica, Glosario de Términos]** para el desarrollo de la aplicación es la 4.4 Kit Kat, por ser la distribución de Android que se encuentra en un mayor número de dispositivos móviles en la actualidad, como se puede observar en la Figura 5, y no tiene grandes exigencias en cuanto a prestaciones hardware, para que pueda ejecutarse de forma fluida por cualquier dispositivo. Asimismo, se ha optado por esta distribución de Android por ciertas mejoras en accesibilidad que facilitarán la experiencia de usuario en terminales móviles, sin que exista un brusco cambio en la interfaz de usuario conocida (ese cambio se hace presente desde la versión 5.0 de Android).

Version	Codename	API	Distribution
2.2	Froyo	8	0.2%
2.3.3 - 2.3.7	Gingerbread	10	4.1%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	3.7%
4.1.x	Jelly Bean	16	12.1%
4.2.x		17	15.2%
4.3		18	4.5%
4.4	KitKat	19	39.2%
5.0	Lollipop	21	15.9%
5.1		22	5.1%

Figura 5: Porcentaje de uso de las versiones de Android presentes en los dispositivos móviles de todo el mundo a fecha de septiembre de 2015.

La barra de estado y la barra de navegación son translúcidas y desaparecen por completo cuando una aplicación se encuentra a pantalla completa. Android 4.4 es compatible con un nuevo gesto, par asegurarse que siempre se tenga acceso a la interfaz de usuario del sistema desde el modo de inmersión a pantalla completa: un toque en los bordes superior e inferior de la pantalla, con el que se desplegará la nueva interfaz del sistema.

En cuanto a otras mejoras de accesibilidad de la distribución 4.4 de Android, se añade un apartado de preferencias para subtítulos, orientado a las personas con discapacidad auditiva, que pueden ser configurados por el usuario en cuanto a idioma, tamaño y estilo de texto. Las aplicaciones pueden acceder a esta configuración y ajustar su presentación según las preferencias del usuario. También se extiende la API de accesibilidad para realizar una

descripción estructural y semántica más precisa, proporcionando acceso a más información sobre los elementos en pantalla.

El listado de opciones que hay en el menú de accesibilidad dispone de dos submenús: servicios y sistemas. En los servicios sólo está incluido Tallback **[Tallback, Glosario de Términos]**, que es el lector de pantallas para Android. En el submenú de sistemas se encuentran estas opciones:

- Subtítulos.
- Gestos de ampliación.
- Texto grande.
- Colgar llamada con botón de encendido.
- Pantalla giratoria.
- Mostrar contraseñas.
- Acceso directo a las opciones de accesibilidad.
- Síntesis de voz.
- Retraso de pulsación prolongada.

Una de las críticas que recibido esta versión de Android en cuanto a opciones de accesibilidad, tiene que ver con los gestos de magnificación. Esta herramienta permite emitir una respuesta la triple pulsación rápida, y también ofrecer una respuesta para al realizar el gesto de acercar o separar los dedos para aumentar o reducir el nivel de zoom que se necesite. Los fallos encontrados ocurren cuando el magnificador de pantalla y el lector de pantalla Tallback están activados al mismo tiempo, y el foco de Tallback está posicionado sobre un elemento concreto, con lo que el magnificador de pantalla se desactiva y sólo se mantiene activado el lector de pantalla. Algunas de estas características de accesibilidad aquí presentadas están relacionadas con ciertos requisitos de la aplicación a desarrollar, como pueden ser el retraso de pulsación prolongada o los gestos de ampliación.

En cuanto a herramientas de apoyo, el sistema operativo Android permite la utilización de productos de apoyo para mejorar la accesibilidad, tales como:

- TextToSpeech: Síntesis de voz que permite al dispositivo móvil convertir texto en audio.
- ExploreByTouch: Funcionalidad que permite al sistema generar eventos de accesibilidad en función de cómo se interaccione con la pantalla del dispositivo, según en que partes de está pulsando con el dedo. A su vez, un lector de pantalla puede

sintetizar y reproducir el contenido y la descripción del elemento por el que se esté navegando en ese momento.

- **Tallback:** Es un lector de pantalla ofrecido por Google en sus dispositivos que lee el elemento que contiene el foco.
- **Tecla Access:** Método de entrada de forma alternativa que facilita el manejo del dispositivo a personas con problemas de movilidad.
- **Soundback:** Genera sonidos al realizar algún acción sobre el dispositivo, como al clicar en una zona de la pantalla. Google lo ofrece siempre en los terminales.
- **Kickback:** Proporciona un feedback de vibración cuando sucede un evento, como puede ser una actualización dinámica de contenido de una aplicación o el desbloqueo de la pantalla del terminal.
- **BrailleBack:** Permite conectar un dispositivo externo de braille para poder escribir y navegar.
- **EyesFree Keyboard:** Ofrece una interfaz para navegar por la pantalla mediante gestos como método de entrada de texto.
- **Dasher:** Método de entrada de texto enfocado para uso de punteros táctiles en la pantalla del dispositivo móvil.

De estos productos de apoyo que se han descrito, la funcionalidad de algunos de ellos está presente en los requisitos de la aplicación, como proporcionar feedback de vibración al recibir una notificación (como hace Kickback) o implementar un control mediante gestos si la interfaz lo permite (como ofrece EyesFree Keyboard).

## 6.2 Análisis: requisitos para la app móvil nativa

Los requisitos de la aplicación nativa de Android se obtendrán a partir de los requisitos del apartado 5. **Requisitos de la aplicación**, que en ese apartado estaban orientados a web móvil. La equivalencia entre requisitos web móvil y nativos se mostrará en la Tabla 14, junto con los identificadores de requisitos web y nativos, el nombre del requisito, cómo se ha implementado y si se esta implementación es total, parcial o no se conseguido resolver.



Requisito Web Móvil	Nombre del requisito	Requisito Nativo	Cómo se ha implementado	Implementación Total/Parcial/No conseguido
RUF-001	Visualizar menús e índices al cargar la página	NAT-001	La API de accesibilidad de Android no ofrece los recursos suficientes.	No conseguido
RUF-002	Permitir controlar la interfaz con un sólo dedo	NAT-002	La API de accesibilidad de Android no ofrece los recursos suficientes.	No conseguido
RUF-003	Autocompletar	NAT-003	La API de accesibilidad de Android no ofrece los recursos suficientes.	No conseguido
RUF-004	Verificar la existencia de un encabezado de página pequeño	NAT-004	La API de accesibilidad de Android no ofrece los recursos suficientes.	No conseguido
RUF-005	Ofrecer acceso rápido a atajos	NAT-005	La API de accesibilidad de Android no ofrece los recursos suficientes.	No conseguido
RUF-006	Permitir uso de métodos abreviados de teclado	NAT-006	La API de accesibilidad de Android no ofrece los recursos suficientes.	No conseguido
RUF-007	Definir área táctil de la pantalla	NAT-007	Mediante los métodos <code>public int getScrollX()</code> y <code>public int getScrollY()</code> , de cualquiera de los tipos de vista de Android.	Parcial
RUF-008	Evitar doble clic	NAT-008	La API de accesibilidad de Android no ofrece los recursos suficientes.	No conseguido



RUF-009	Permitir pulsación prolongada	NAT-009	Mediante el atributo public static final int TYPE_VIEW_LONG_CLICKED, que representa una pulsación prolongada, y el método void setLongClickable (boolean pulsacionProlongada), establece si se puede hacer una pulsación prolongada.	Parcail
RUF-010	Permitir alternativas para pulsaciones simultáneas de teclas	NAT-010	La API de accesibilidad de Android no ofrece los recursos suficientes.	No conseguido
RUF-011	Usar los botones de control físicos para navegar por la interfaz de la aplicación	NAT-011	La API de accesibilidad de Android no ofrece los recursos suficientes.	No conseguido
RUF-012	Implementar un control mediante gestos si la interfaz lo permite	NAT-012	La clase de Java AccessibilityService dispone de atributos que representan a cada tipo de evento, entre los que se encuentran los relacionados con el control de gestos. Para obtener el tipo de evento se usa el metodo getEventType(), así como los atributos int TYPE_TOUCH_EXPLORATION_GESTURE_START, int TYPE_TOUCH_EXPLORATION_GESTURE_END, int TYPE_TOUCH_GESTURE_DETECTION_START, int TYPE_TOUCH_GESTURE_DETECTION_END.	No conseguido

RUF-013	Incluir enlace o botón de complemento para la funcionalidad	NAT-013	La API de accesibilidad de Android no ofrece los recursos suficientes.	No conseguido
RUF-014	Evitar ventanas emergentes o pop-ups	NAT-014	Mediante la clase Java AccessibilityNodeInfo, método boolean canOpenPopUp(), que permite comprobar si el elemento actual se abre mediante una ventana emergente o un cuadro de diálogo. Hay que tener en cuenta también el atributo TYPE_WINDOW_STATE_CHANGED, que representa la apertura de una ventana emergente.  Con el método void setOpenPopUp(boolean opensPopUp) de la clase AccessibilityNodeInfo, si se indica FALSE en el parámetro del método, se consigue evitar abrir ventanas emergentes o pop-ups.	Total
RUF-015	Adaptar el teclado virtual al contenido que se va a introducir	NAT-015	Mediante la clase KeyBoardView, que es la que se encarga de pintar en pantalla la interfaz del InputMethod. Deberá tener en cuenta donde está posicionado el foco.	Parcial
RUF-016	Proporcionar feedback al usuario mediante un led de color	NAT-016	Mediante el atributo FEEDBACK_VISUAL de la clase AccessibilityEvent, y el método void sendAccessibilityEvent(AccessibilityEvent	Total

			evento), que contendrá como parámetro el evento con el feedback de respuesta.	
RUF-017	Proporcionar feedback al usuario mediante un sonido	NAT-017	Mediante el atributo <code>FEEDBACK_AUDIBLE</code> de la clase <code>AccessibilityEvent</code> , y el método <code>void sendAccessibilityEvent(AccessibilityEvent evento)</code> , que contendrá como parámetro el evento con el feedback de respuesta.	Total
RUF-018	Proporcionar feedback al usuario mediante una vibración	NAT-018	Mediante el atributo <code>FEEDBACK_HAPTIC</code> de la clase <code>AccessibilityEvent</code> , y el método <code>void sendAccessibilityEvent(AccessibilityEvent evento)</code> , que contendrá como parámetro el evento con el feedback de respuesta.	Total
RUF-019	Configurar el tipo de respuesta que se ofrecerá como feedback al usuario	NAT-019	La API de accesibilidad de Android no ofrece los recursos suficientes.	No conseguido
RUF-020	Ofrecer información de estado al usuario	NAT-020	Mediante los métodos <code>int getWindowId()</code> , <code>AccessibilityNodeInfo getSource()</code> y el atributo <code>String CATEGORY_STATUS</code> de la clase <code>Notification</code> .	Parcial
RUF-021	Gestionar notificación de aviso	NAT-021	Mediante las clases <code>Notification</code> y <code>NotificationManager</code> , y el atributo <code>String CATEGORY_MESSAGE</code> .	Parcial

RUF-022	Gestionar notificación de error	NAT-022	Mediante las clases Notification y NotificationManager, y el atributo String CATEGORY_ERROR.	Parcial
RUF-023	Gestionar notificación de alerta	NAT-023	La API de accesibilidad de Android no ofrece los recursos suficientes.	No conseguido
RUF-024	Identificar elementos de la interfaz	NAT-024	Rellenar el campo ContentDescription en los elementos gráficos ImageView, ImageButton...etc.	No conseguido
RUF-025	Pausar o detener la presentación de información	NAT-025	La API de accesibilidad de Android no ofrece los recursos suficientes.	No conseguido
RUF-026	Controlar el tiempo de presentación de información	NAT-026	La API de accesibilidad de Android no ofrece los recursos suficientes.	No conseguido
RUF-027	Ofrecer alternativas a imágenes animadas	NAT-027	La API de accesibilidad de Android no ofrece los recursos suficientes.	No conseguido
RUF-028	Acceder con el tabulador a un elemento de la página	NAT-028	La API de accesibilidad de Android no ofrece los recursos suficientes.	No conseguido
RUF-029	Incluir información sobre el rol o función de un elemento de la página	NAT-029	Mediante el método void setContentDescription(CharSequence info) de la clase AccessibilityNodeInfo.	Parcial
RUF-030	Permitir a un elemento dinámico cambiar de estado	NAT-030	Mediante el atributo TYPE_WINDOW_CONTENT_CHANGED, que permite a un elemento dinámico cambiar de	Parcial

			estado, en concreto el subárbol de la raíz del evento.	
RUF-031	Identificar una zona dinámica del contenido que se actualiza automáticamente	NAT-031	Mediante la clase Java AccessibilityNodeInfo, método int getLiveRegion(), que permite obtener una zona dinámica del contenido que se actualiza de forma automática.	Total
RUF-032	Etiquetar un elemento	NAT-032	Mediante la clase Java AccessibilityNodeInfo, método AccessibilityNodeInfo getLabeledBy(), que obtiene la información que permite etiquetar un elemento	Parcial
RUF-033	Mejorar la accesibilidad de los campos de formularios	NAT-033	La API de accesibilidad de Android no ofrece los recursos suficientes.	No conseguido
RUF-034	Ocultar elementos a los productos de apoyo	NAT-034	La API de accesibilidad de Android no ofrece los recursos suficientes.	No conseguido
RUF-035	Definir un estado nuevo para los checkbox	NAT-035	La API de accesibilidad de Android no ofrece los recursos suficientes.	No conseguido
RUF-036	Verificar que el rol de un elemento dinámico es fijo	NAT-036	La API de accesibilidad de Android no ofrece los recursos suficientes.	No conseguido
RUF-037	Usar marcado nativo cuando sea posible	NAT-037	La API de accesibilidad de Android no ofrece los recursos suficientes.	No conseguido
RUF-038	Marcar las relaciones entre los elementos con el atributo más	NAT-038	La API de accesibilidad de Android no ofrece	No conseguido



	apropiado		los recursos suficientes.	
RUF-039	Sincronizar la interfaz visual con la interfaz accesible	NAT-039	Hay que apuntar al componente que quiera mover el foco en cada una de las 4 direcciones de navegación, mediante el método <code>isFocusable()</code> , ej: <code>View v = findViewById(R.id.myView); if(!v.isFocusable()){ v.setFocusable(true); } if(!v.isFocusableInTouchMode()){ v.setFocusableInTouchMode(true); }</code>	No conseguido
RUF-040	Proporcionar mecanismos redundantes de interacción para el manejo de scripts dinámicos	NAT-040	La API de accesibilidad de Android no ofrece los recursos suficientes.	No conseguido

Tabla 14: Equivalencia de requisitos entre una aplicación web orientada a móvil y una app nativa

## 6.3 Diseño: interfaz de usuario, funcionalidad, controlador y diagrama de clases

### INTERFAZ DE USUARIO

La interfaz de usuario se compone de 3 pantallas (3 actividades).

La actividad principal tendrá un botón para cargar el fichero de la sencilla aplicación que va a analizar la app de accesibilidad (la aplicación de evaluación de accesibilidad del proyecto analiza una pequeña aplicación Android). Para no complicar la manera de hacerlo se abrirá un fichero con el código fuente de la app a analizar, en lugar de coger el archivo ejecutable .apk de la app, descompilarlo y analizar el código fuente. Esta actividad también tendrá un botón que comenzará el análisis de evaluación de accesibilidad de la app.

La segunda actividad muestra los resultados del análisis de la app mediante un informe de evaluación de accesibilidad desglosado en requisitos de accesibilidad que se cumplen totalmente, parcialmente, no se cumplen o no se detectan, y avisos o alertas de mal empleo de accesibilidad en alguna parte de los elementos de la interfaz de usuario de la app analizada.

La tercera actividad muestra un mensaje de error en caso de no abrirse correctamente el fichero de la app a analizar o cualquier otra excepción que ocurra.

### FUNCIONALIDAD

La aplicación de evaluación de accesibilidad analizará un app nativa Android sencilla, siempre que sea tenga problemas en abrir el fichero de la misma, y analizará su código en busca de los requisitos de accesibilidad propuestos en este proyecto. Una vez acabado el análisis, mostrará el resultado en otra pantalla dividido en: requisitos de accesibilidad cumplidos, alertas, recomendaciones y requisitos de accesibilidad no cumplidos. Cualquier error o anomalía que se produzca antes o durante el análisis, sea en la apertura del archivo de la app a analizar, o durante el análisis de la misma, se interrumpirá la ejecución de la aplicación para redirigirse a una pantalla indicando el motivo de la parada o el problema detectado.

### CONTROLADOR

Una Activity [**Activity, Glosario de Términos**] representa cada pantalla y la interacción con el usuario. ésta se relacionará con un Service [**Service, Glosario de Términos**], que es una operación de larga duración en segundo plano. También hay que realizar la suscripción al servicio. Con un Intent [**Intent, Glosario de Términos**] podremos mostrar una actividad desde cualquier otra, iniciar un servicio o mandar un mensaje, como se puede observar a modo de resumen en las figuras 6 y 7.

También hay que tener en cuenta la captura y gestión de eventos, como pueden ser notificaciones, pulsaciones de teclas del teclado virtual, gestos, actualizaciones dinámicas...etc. Estos eventos hay que tratarlos según la API de accesibilidad de Android, por lo que se usarán métodos apropiados para ello, como pueden ser: `sendAccessibilityEvent()` o `dispatchPopulateAccessibilityEvent()`.

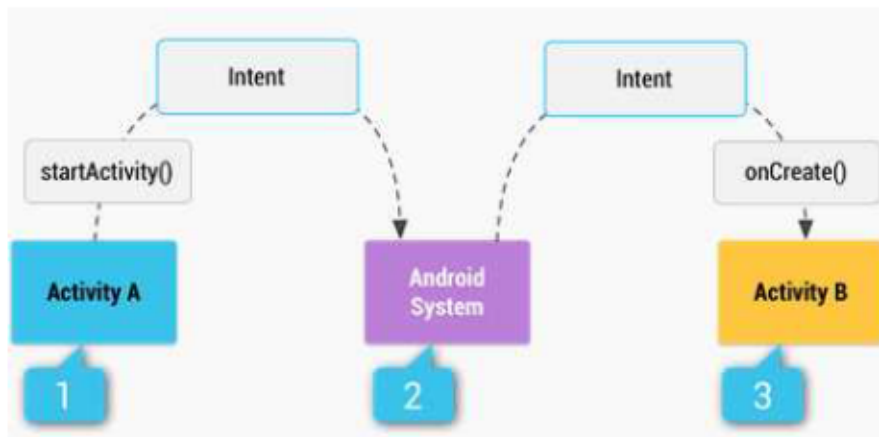


Figura 6: Relación entre Activity e Intent en Android.

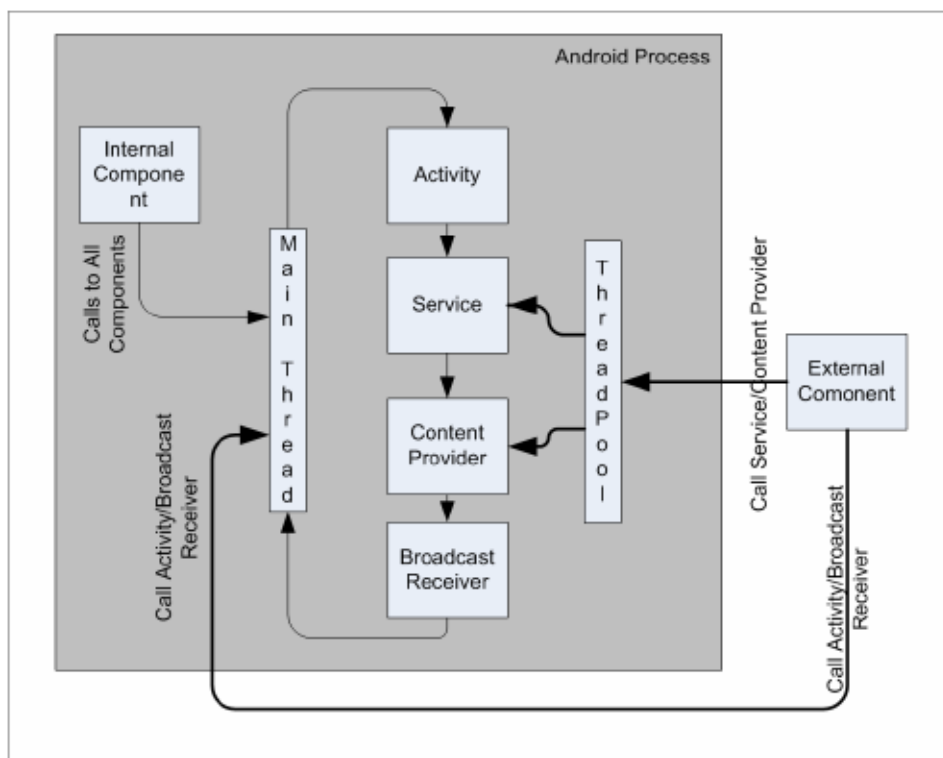
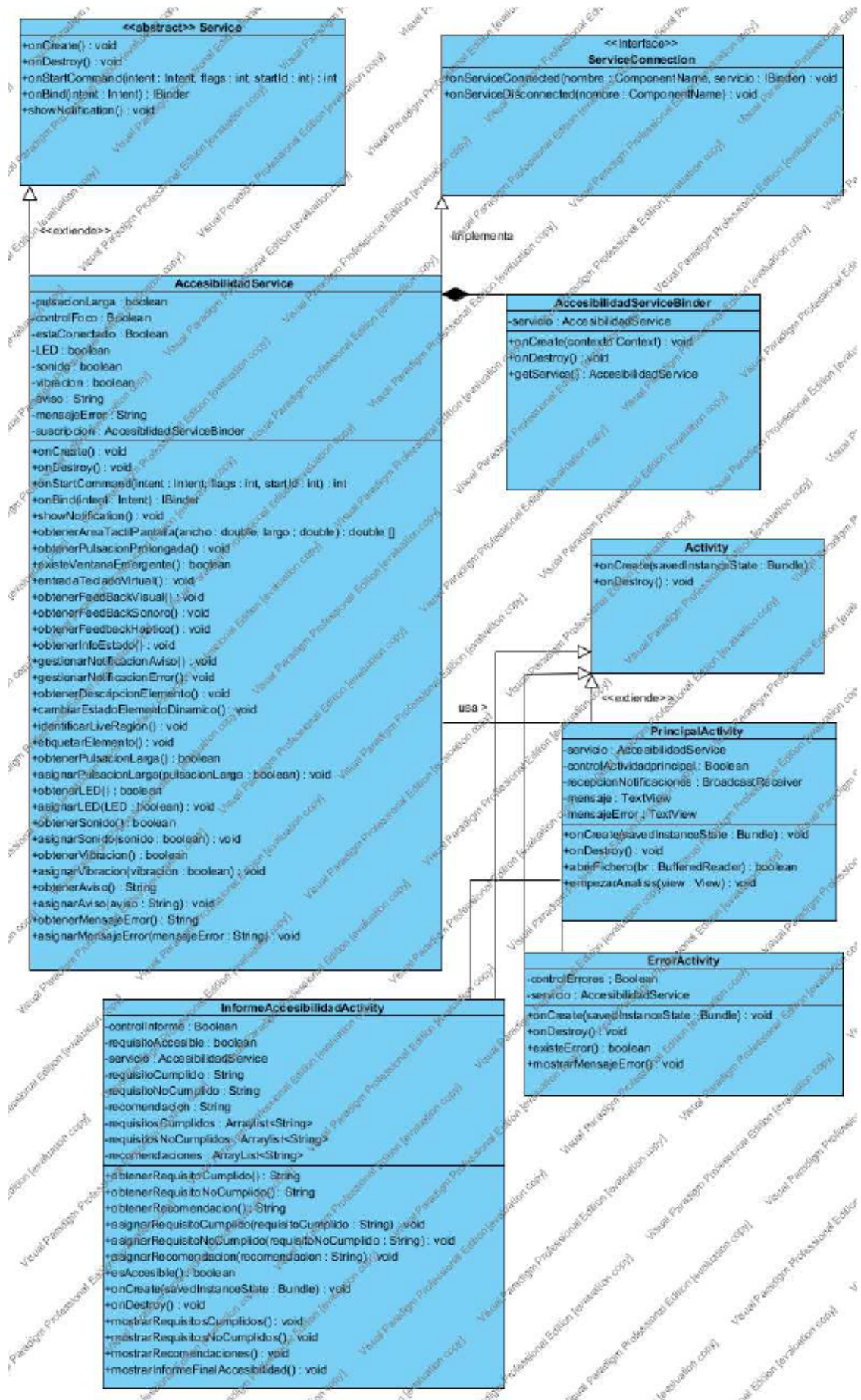


Figura 7: Relación entre Activity y Service en Android.

## DIAGRAMA DE CLASES

La jerarquía de clases Java que se usarán para el desarrollo de la aplicación es el siguiente:





## 6.4 Alternativas de diseño

Existen algunos framework que permiten el desarrollo de aplicaciones híbridas, basadas en HTML5 y Javascript, y con apariencia similar a las nativas de cada plataforma móvil.

**PhoneGap** es un framework **[Framework, Glosario de Términos]** gratuito y de código abierto que permite la creación de aplicaciones móviles multiplataforma usando tecnologías como HTML5, CSS3 y Javascript. Se usa para crear aplicaciones híbridas, a mitad de camino entre las nativas y las que son puramente web, mediante unas APIs que permiten controlar las características del dispositivo. Las aplicaciones que se crean con PhoneGap sólo pueden usar HTML, CSS y Javascript. Si estas apps requieren de código en otros lenguajes de programación, debe ser mediante un backend **[Backend, Glosario de Términos]** a través de APIs o webservices.

Dispone de una librería Javascript y otra en el lenguaje nativo que usa la plataforma para desarrollar aplicaciones, de forma que tiene APIs similares a las nativas de Android pero facilitando el desarrollo mediante la programación con HTML5 y Javascript.

**Sencha Touch** es un framework que utiliza los estándares web de HTML5, CSS3 y Javascript para desarrollar aplicaciones web para dispositivos móviles con la apariencia de ser aplicaciones nativas de los sistemas iOS, Android y Blackberry. Proporciona gran cantidad de eventos táctiles, incluso permite convertir la aplicación híbrida desarrollada en nativa.

Otro ejemplo de framework, basado en el lenguaje de programación Ruby, es **Rhodes**, que permite construir aplicaciones nativas para los principales sistemas operativos móviles.

Las aplicaciones híbridas tienen mayor facilidad de desarrollo y mantenimiento de la aplicación, sin embargo, no alcanza tanto rendimiento como una aplicación nativa. En este proyecto se ha decidido realizarlo sobre aplicaciones nativas al no existir apenas software accesible de este tipo, ya que hay gran abundancia de aplicaciones móviles basadas en web, o incluso aplicaciones nativas basadas en HTML5.

## 6.5 Implementación y arquitectura de la aplicación

La arquitectura de Android está constituida por varias capas que permiten al desarrollador la creación de aplicaciones, como se puede observar en la Figura 6. Esta organización en capas permite al desarrollador acceder a las capas más bajas, más cercanas al hardware del dispositivo, mediante el uso de librerías de forma que no tenga que programar a bajo nivel con algún tipo de lenguaje ensamblador, de esta forma la aplicación puede hacer uso de los componentes hardware del dispositivo.



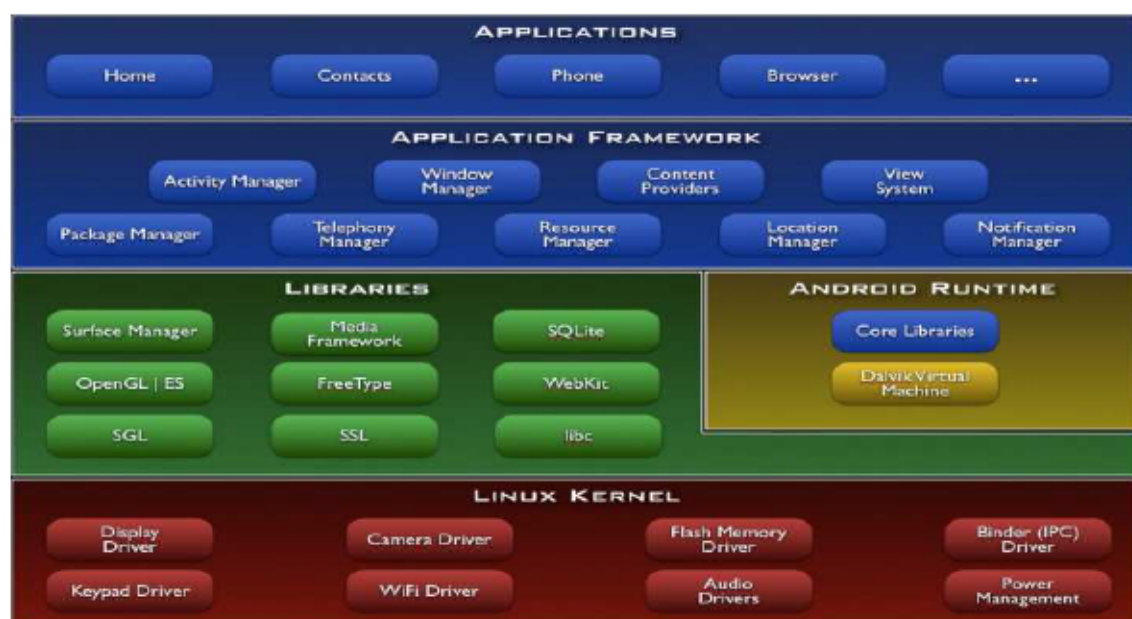


Figura 6: Capas de la arquitectura de Android.

Cada una de las capas utiliza o tiene acceso a los elementos de la capa inferior para realizar sus funciones en un tipo de arquitectura denominada pila **[Pila, Glosario de Términos]**. Estas capas son:

- Kernel de Linux:** El kernel es el núcleo del sistema operativo, la versión 4.4 de Android está basado en kernel Linux 2.6 (la versión 5.0 de Android dispone del kernel Linux más reciente, la versión 3.14). El núcleo ofrece una capa de abstracción entre el hardware y las demás capas de la arquitectura. Para componente hardware del dispositivo, existe un driver o controlador **[Driver, Glosario de Términos]** dentro del kernel que permite utilizarlo mediante una llamada al sistema **[Llamada al Sistema, Glosario de Términos]** (mediante software). El kernel también se encarga de gestionar diferentes recursos del sistema operativo (memoria, temperatura del dispositivo, procesos en ejecución...).
- Librerías:** La capa que está encima del kernel es la que contiene las bibliotecas nativas de Android (librerías del sistema) **[Librerías del Sistema, Glosario de Términos]**. Están programadas en C o C++, y compiladas para la arquitectura hardware específica del dispositivo. Estas librerías ofrecen funcionalidad a las aplicaciones para tareas que se repiten con mucha frecuencia, reutilizando código de forma óptima para que se consiga evitar que se codifiquen cada vez. Existen muchas librerías en Android, tales como: SQLite (gestión de base de datos), OpenGL (motor gráfico), Webkit (navegador), Media Framework (soporte de codecs de reproducción y grabación de audio y vídeo), Surface Manager (acceso al subsistema de representación gráfica 2D y 3D), SGL (motor

de gráficos 2D), FreeType (fuentes en bitmap y renderizado vectorial) [**Bitmap, Glosario de Términos**] [**Renderizado Vectorial, Glosario de Términos**] y SSL (servicios de encriptación Secure Socket Layer) [**SSL, Glosario de Términos**].

- **Entorno de ejecución** (Android Runtime): Está formado por librerías específicas de Android y otras librerías de lenguaje de programación Java [**Lenguaje de Programación Java, Glosario de Términos**]. El componente principal del entorno de ejecución de Android 4.4 es la máquina virtual Dalvik, por defecto, aunque para esta versión de Android también se ha introducido una máquina virtual opcional para los desarrolladores, ART, con el objetivo del ahorro energético en mente. Para este proyecto se usará la máquina virtual Dalvik. Las aplicaciones de Android se programan principalmente en Java (existen otros lenguajes como Go con el que también es posible), y son compiladas en un formato específico, que no es el bytecode [**Bytecode, Glosario de Términos**] que genera Java al compilar, sino un ejecutable específico de la máquina virtual Dalvik, con extensión .dex, de forma que no es posible ejecutar aplicaciones Java en Android ni aplicaciones Android en Java. De esta forma las aplicaciones se compilan sólo una vez y se pueden ejecutar en cualquier dispositivo Android que tenga la versión mínima del sistema operativo que requiera la aplicación.
- **Framework** [**Framework, Glosario de Términos**] **de aplicaciones:** Esta capa está formada por la totalidad de las clases y servicios que utilizan directamente las aplicaciones para realizar sus funciones. La mayor parte de los componentes de esta capa suelen ser librerías Java que acceden a los recursos de las capas inferiores a través de la máquina virtual Dalvik. Son las siguientes:
  - Activity Manager: Administra la pila de actividades de la aplicación y su ciclo de vida.
  - Windows Manager: Organiza lo que se mostrará en pantalla, lo que serán las actividades.
  - Views: Son las vistas con los elementos que permiten construir las interfaces de usuario: listas, botones, navegador web...etc.
  - Notification Manager: Administra los servicios que notifican al usuario cuando se requiera su atención mostrando alertas en la barra de estado. Para las notificaciones permite usar leds de colores, sonido o vibración, siempre que lo permita el hardware del dispositivo.
  - Package Manager: Permite obtener información sobre los paquetes instalados en el dispositivo Android y gestionar la instalación de nuevos

paquetes. Esta es la forma en la que se distribuyen las aplicaciones Android, conteniendo estos paquetes el archivo ejecutable de la aplicación Android .apk, y estos a su vez contienen los archivos .dex con todos los recursos y archivos adicionales que necesite la aplicación, facilitando su descarga e instalación.

- **Resource Manager**: Gestiona todos los elementos que forma parte de la aplicación que están fuera del código (imágenes, sonidos, traducción de texto a distintos idiomas...etc).

- **Location Manager**: Determina la posición geográfica del dispositivo Android mediante posicionamiento GPS u otras redes.

- **Content Provider**: Crea una capa que encapsula los datos que se van a compartir entre aplicaciones para tener el control sobre cómo acceder a la información.

- **Sensor Manager**: Controla los elementos del hardware del dispositivo (sensor de temperatura, sensor de luminosidad, giroscopio, acelerómetro, sensor de proximidad...etc.).

- **Telephony Manager**: Administra las llamadas y el envío/recepción de los mensajes SMS/MMS, sin permitir eliminar la actividad mostrada cuando una llamada está en curso.

- **Multimedia**: Reproducción y visualización de imágenes, audio y vídeo en el dispositivo.

- **Cámara**: Permite hacer uso de la API de la cámara para realizar fotografías o grabación de vídeo.

En la aplicación desarrollada se han utilizado como Framework de aplicaciones más importantes los siguientes: Activity Manager, Notification Manager, Package Manager, Resource Manager, Content Provider y Sensor Manager.

- **Aplicaciones**: La última capa incluye todas las aplicaciones del dispositivo: las que tienen interfaz de usuario, las programadas en C o C++, las administradas, las preinstaladas en el dispositivo y las que el usuario instala por su cuenta. La aplicación principal del sistema Home (Inicio) permite ejecutar otras aplicaciones o widgets **[Widget, Glosario de Términos]** mediante una lista y coloca los accesos directos de las mismas en las distintas pantallas del escritorio del sistema.

Para el desarrollo de la aplicación se ha utilizado el entorno de desarrollo Eclipse 4.4 (Luna) con el plugin ADT indicado para Android 4.4, y con la API número 19, que es la pertenece a esta versión de Android. Se ha usado también el JDK 8 de Java para programar.



Se ha ido estudiando las posibilidades que ofrece los distintos elementos de la API de Android, desde los más básicos, como Activity [Activity, Glosario de Términos], Service [Service, Glosario de Términos] o Intent [Intent, Glosario de Términos], hasta los permisos requeridos por la aplicación Android, como puede ser conexión a Internet, escritura...incluso como se organiza la seguridad de una aplicación Android mediante aislamiento de procesos, llamado sandboxing [Sandboxing, Glosario de Términos].

## 6.6 Verificación y pruebas de la herramienta de evaluación

En este subapartado se realizará una batería de pruebas de la aplicación desarrollada, como resultados de la ejecución de datos de prueba:

**Prueba 1:** No hay fichero de app nativa para analizar.

Resultado: Salta a la actividad de error de la aplicación con un mensaje que indica que no se pudo abrir el fichero.

**Prueba 2:** Fichero de la app nativa con otra extensión.

Resultado: Salta a la actividad de error de la aplicación con un mensaje que indica que no es la extensión correcta de una app nativa.

**Prueba 3:** Fichero de la app nativa con una extensión correcta.

Resultado: Ejecuta la aplicación normalmente.

**Prueba 4:** Fichero de la app nativa con apenas código que analizar.

Resultado: Ejecuta la aplicación normalmente, aunque al no tener apenas código la mayoría de los requisitos de accesibilidad no se cumplirán.

**Prueba 5:** Fichero de la app nativa con gran cantidad de código a analizar.

Resultado: Ejecuta la aplicación normalmente y muestra el informe de resultados. En alguna ejecución aislada, si la app nativa a analizar tenía demasiadas líneas de código, la ejecución no ha llegado a obtener el informe final de accesibilidad, o lo ha hecho al cabo de mucho tiempo.

**Prueba 6:** Introducción de caracteres, números en la apertura de fichero.

Resultado: Salta a la actividad de error de la aplicación con un mensaje que indica que no es la extensión correcta de una app nativa.

**Prueba 7:** Durante el análisis de la app nativa, se interacciona de forma intensa con la actividad principal (la pantalla) de la app desarrollada.

Resultado: En algún casos se ha llegado a parar bruscamente la aplicación sin motivo aparente.

Respecto a la aplicación AMP Mobile [AMP MOBILE, "<http://www.ssbartgroup.com/blog/amp-for-mobile-is-here/>"], al permitir realizar pruebas en el mismo dispositivo móvil o mediante un emulador, aseguran un nivel óptimo de precisión, que permite a los desarrolladores realizar pruebas iterativas y de forma rápida durante la fase

de desarrollo en lugar de analizar directamente el código, con el objetivo de identificar y solucionar problemas de accesibilidad durante el proceso de desarrollo real. Según los desarrolladores de esta herramienta, es más eficiente realizar pruebas de accesibilidad en las primeras etapas de desarrollo que no hacerlo posteriormente, y ese es un punto de vista que confronta con lo que se propone en este proyecto, ya que se pretende realizar una aplicación nativa Android que analice otra app nativa en su estado final, no en las primeras fases de desarrollo, demostrando que no se pierde eficiencia analizando una aplicación nativa Android de esta forma.



## 7. Conclusiones y trabajos futuros

### 7.1 Conclusiones, principales aportaciones y consecución de objetivos

El estudio y desarrollo de este Trabajo de Fin de Grado ha permitido concienciarse de la importancia que debe tener el acceso a los recursos digitales hoy en día, especialmente en los dispositivos móviles, donde siempre agradecerán los usuarios, ya sean discapacitados, con alguna limitación temporal e incluso las personas mayores, que las aplicaciones móviles sean lo más accesibles que se pueda, además de usables.

Gracias a la contribución de muchas organizaciones con sus respectivas guías de evaluación de accesibilidad, han tratado de ir mejorando la experiencia de usuario de todas las personas. Para ello, las diferentes leyes, reales decretos, normativas y estándares de accesibilidad han facilitado ese camino hasta llegar donde estamos hoy.

No sólo es una responsabilidad social sino que también permite abrir la cuota de mercado y la audiencia mejorando el acceso a los contenidos web y móviles, reduciendo costes de desarrollo y mantenimiento de distintas aplicaciones e incrementando el soporte para el mercado internacional, mediante la ayuda de subtítulos, idiomas alternativos y contenidos universales.

Al encontrarnos un mercado actual que no trata apenas la accesibilidad móvil en aplicaciones nativas, ni dedican tiempo y esfuerzo al desarrollo de herramientas de evaluación de accesibilidad para las mismas, este proyecto ha sido una respuesta a ese problema, tomándolo con motivación y dando un nuevo impulso a este campo de la informática.

El seguimiento del proyecto ha ayudado a la mejora del mismo, e incluso ha agilizado su desarrollo.

Los principales beneficios aprendidos con la realización de este proyecto son:

- Conseguir que una aplicación o página web sea más visitada al no presentar barreras para su acceso, independientemente de las condiciones del usuario, y siguiendo el cumplimiento de los estándares de accesibilidad, visualizarse correctamente en cualquier dispositivo.
- Reducir tiempo de carga de aplicaciones y páginas web al separar el contenido de la información de la capa de presentación. Esto también fomenta el correcto mantenimiento de la aplicación móvil o web, sencillez para actualizar su contenido y contener menos errores.
- Un aumento en la accesibilidad de una aplicación móvil o web implica una mayor usabilidad de la misma.

- Potenciar la mejora del acceso a los recursos entre todos, dado que la Administración Pública fomenta cualquier petición de mejora de accesibilidad de sus ciudadanos.
- Soltura con la plataforma Android y con su distribución 4.4.
- Los desarrollos nativos hacen uso del lenguaje nativo del sistema obteniendo aplicaciones con un rendimiento óptimo.

Durante el desarrollo de este proyecto, no sólo he sido consciente de los pasos que iba alcanzando en materia de accesibilidad, sino también de que mi pequeña aportación pueda servir a quien quiera adentrarse en el mundo de la accesibilidad, tanto web como especialmente en dispositivos móviles, con el objetivo común de que, entre todos, consigamos la brecha digital que a veces existe en la sociedad.

## 7.2 Dificultades encontradas

Al inicio del proyecto, al no tener los conocimientos necesarios sobre accesibilidad web, era difícil distinguir entre toda la nomenclatura existente, como normativas, estándares, guías de accesibilidad, recomendaciones, metodologías...todo parecía lo mismo descrito de distinta forma, hasta ya ir adquiriendo más soltura con esos términos.

La planificación del proyecto final refleja que las tareas asociadas a cada fase requerían de mucho más tiempo del inicialmente previsto, por lo que mediante seguimiento se tuvo que ir reorientando los objetivos en cada momento.

El estudio de las distintas guías de accesibilidad de las distintas organizaciones requirió de un gran esfuerzo por intentar aunarlas de forma que se pudieran utilizar de alguna forma en la propuesta de guía de evaluación de accesibilidad, y posteriormente obtener de esas pautas de accesibilidad los requisitos de la aplicación, que si ya tuvo su pequeña dificultad para tratarlo como contenido web, a la hora de convertirlo a contenido nativo, fue muy costoso en tiempo y dedicación, ya que apenas hay documentación y únicamente uno se puede orientar con las breves descripciones de la librería de accesibilidad de Android, y en algunos momentos, la captura y gestión de eventos se hizo muy compleja para conseguir obtener los resultados de accesibilidad propuestos en los requisitos.

## 7.3 Trabajos futuros

Este Trabajo de Fin de Grado abarca mucho más contenido del realmente desarrollado en la aplicación, ya que en la propuesta de guía de evaluación de accesibilidad hay muchas categorías, subcategorías y pautas de accesibilidad que no se tratan en profundidad a la hora de desarrollar la aplicación nativa en Android, por lo que alguien interesado en la materia puede adentrarse en el resto de categorías de accesibilidad definidas en la guía de evaluación de accesibilidad, para el posterior desarrollo de las mismas en otra aplicación nativa, o ampliando la que se desarrolla en este proyecto.

Para realizar una buena evaluación de accesibilidad de la aplicación desarrollada, hay que cumplir cierta metodología y una serie de pasos a seguir en el futuro:

1. Herramienta automática de evaluación de accesibilidad, que se encargue de analizar el código de la aplicación nativa para comprobar qué grado o nivel de accesibilidad tiene la aplicación desarrollada según las herramientas de evaluación actuales.
2. Evaluación de un experto en accesibilidad, que compruebe el seguimiento y cumplimiento de las pautas de la propuesta de guía de accesibilidad y cómo se implementan en la aplicación nativa.
3. Evaluación de los usuarios, que los que realmente pueden examinar una aplicación con su uso frecuente, para aportar soluciones a fallos encontrados o sugerencias de mejoras en la funcionalidad de la aplicación.

## 8. Glosario de términos

**Accesibilidad:** Posibilidad de que un producto o servicio web pueda ser usado por el mayor número posible de personas, independientemente de las limitaciones propias del individuo o de las derivadas del contexto de uso. Acceso universal a los recursos independientemente del tipo de hardware, software, infraestructura de red, idioma, cultura, localización geográfica o capacidades de los usuarios.

**Activity:** Cada pantalla de la aplicación. Permite interacción con el usuario.

**AMP Mobile:** Plataforma de Gestión de Accesibilidad para aplicaciones nativas móviles y también tiene una versión para contenido web. Es un nuevo motor de pruebas para el entorno móvil en el que se realiza una prueba automatizada de la accesibilidad de dispositivos móviles directamente desde los entornos de desarrollo móvil y de forma ágil. Identifica y resuelve violaciones de accesibilidad a través de pruebas rápidas de forma iterativa en las primeras etapas de desarrollo, en lugar de hacer un simple análisis de código fuente.

**API (Interfaz de Programación de Aplicaciones):** Conjunto de subrutinas, funciones y procedimientos que ofrece una biblioteca de desarrollo para ser utilizado otro software como una capa de abstracción.

**App Móvil:** Aplicación informática diseñada para ser ejecutada en teléfonos inteligentes, tabletas y otros dispositivos móviles.

**Backend:** Administración de un sitio web desde el lado del servidor.

**BBC:** organización que incorpora una serie de estándares promovidos por la W3C para facilitar el acceso a los contenidos publicados en sitios web a aquellos usuarios que utilizan dispositivos especiales, como lectores de pantalla.

**Bytecode:** Código intermedio más abstracto que el código máquina.

**Bitmap:** Estructura o fichero de datos que representa una rejilla rectangular de píxeles o puntos de color, llamada matriz, que se puede visualizar en un monitor, papel u otro dispositivo de representación.

**Captcha:** Test controlado por una máquina que realiza una prueba o desafío utilizada en computación, a modo de respuesta para determinar cuando el usuario es o no humano.

**Cookie:** Pequeña información enviada por un sitio web y almacenada en el navegador del usuario, de forma que el sitio web puede consultar la actividad previa del usuario.

**CPU:** Unidad Central de Procesamiento. Hardware dentro de una computadora u otros dispositivos programables que interpreta las instrucciones de un programa informático mediante la realización de las operaciones básicas aritméticas, lógicas y de entrada/salida del sistema.



**Cuello de botella:** La capacidad de procesamiento de un dispositivo es mayor que la capacidad del bus al que se encuentra conectado el dispositivo.

**Deque Systems:** Empresa que contribuye a que las personas con discapacidad puedan disfrutar plenamente de la web

**Deque Systems Axe:** Herramienta de accesibilidad web publicada recientemente que realiza una prueba de accesibilidad automatizada, teniendo en cuenta el navegador y el tipo de pruebas a realizar. Se ejecuta en el servidor de desarrollo local en el mismo navegador, de forma que su ejecución hace la prueba de accesibilidad rápida, confidencial y segura, a diferencia de otros servicios web.

**Diagrama WBS o EDT:** Descomposición jerárquica orienta a crear los entregables requeridos en un proyecto de forma simple y organizada.

**Driver:** Controlador de dispositivo. Parte del sistema operativo encargada de la interacción con todos los controladores.

**Experiencia de usuario:** Conjunto de factores y elementos relativos a la interacción del usuario, con un entorno o dispositivo concretos, cuyo resultado es la generación de una percepción positiva o negativa de dicho servicio, producto o dispositivo.

**Framework:** Infraestructura Digital. Estructura conceptual y tecnológica de soporte definido, con módulos de software concretos, que sirve de base para la organización y desarrollo de software.

**Funka Nu:** Organización de Suecia que fue desarrollando como proyecto unas pautas para el desarrollo de interfaces móviles accesibles, dependiente de la Fundación para Infraestructuras de Internet.

**HTML:** Lenguaje de marcado y estándar que sirve de referencia para la elaboración de páginas web.

**Intent:** Elemento básico de comunicación entre los distintos componentes Android. Son como los mensajes o peticiones que son enviados entre los distintos componentes de una aplicación o entre distintas aplicaciones. Esto permite mostrar una actividad desde cualquier otra, iniciar un servicio u otra aplicación o enviar mensajes.

**Lenguaje de Programación Java:** Lenguaje de programación orientado a objetos que permite a los desarrolladores de aplicaciones escribir el programa una vez y ejecutarlo en cualquier dispositivo, ya que el código no tiene que volver a ser recompilado en cada nuevo dispositivo.

**Librerías del Sistema:** Biblioteca. Conjunto de implementaciones funcionales codificadas en un lenguaje de programación, que ofrece una interfaz bien definida para la funcionalidad que se invoca.

**Llamada al Sistema:** Evento de solicitud del servicio del sistema operativo. Los programas de usuario acceden a los servicios del sistema operativo a través de las llamadas al sistema.



**Memoria caché:** Memoria de acceso rápido de una computadora que guarda temporalmente los datos recién procesados.

**Memoria RAM:** Memoria de Acceso Aleatorio. Memoria de trabajo de computadoras para el sistema operativo, los programas y la mayor parte del software.

**Metodología Ágil de Desarrollo:** Metodología de desarrollo que promueve las reuniones en vez de la documentación, con el objetivo de minimizar los riesgos desarrollando software en lapsos cortos de tiempo. Esos lapsos constituyen una iteración según esta metodología y debe durar entre una y cuatro semanas. Una iteración no debe incluir demasiada funcionalidad para justificar el lanzamiento del producto al mercado, sino que lo que se pretende es conseguir una demo de la aplicación al final de cada iteración, y en ese momento se vuelven a evaluar las prioridades del proyecto para decidir por donde avanzar y que recursos optimizar.

**Node.js:** Entorno en tiempo de ejecución multiplataforma para la capa del servidor, basado en el lenguaje de programación ECMAScript. Es de código abierto, asíncrono, orientado a objetos y gestiona entrada/salida de datos.

**Phonegap:** Framework gratuito y de código abierto que permite la creación de aplicaciones móviles multiplataforma usando tecnologías como HTML5, CSS3 y Javascript. Se usa para crear aplicaciones híbridas, a mitad de camino entre las nativas y las que son puramente web, mediante unas APIs que permiten controlar las características del dispositivo. Las aplicaciones que se crean con PhoneGap sólo pueden usar HTML, CSS y Javascript.

**Pila:** Estructura de datos en la que el modo de acceso a sus elementos es de tipo LIFO (Last-In-First-Out, último en entrar-primero en salir) que permite almacenar y recuperar datos.

**Plataforma Tecnológica:** Conjunto de aplicaciones, componentes software, técnicas, lenguajes de programación y servicios, que se utilizan para el desarrollo de las aplicaciones empresariales.

**Productos de Apoyo:** Dispositivos empleados por las personas con discapacidad para atenuar, prevenir o neutralizar esa discapacidad. Hay diversas tecnologías de apoyo o ayudas técnicas que pueden usar los usuarios discapacitados para usar un dispositivo móvil o navegar por una página web, tales como un dispositivo hardware que convierte texto en caracteres Braille (llamado "Líneas Braille"); un lector de pantalla que disponga de síntesis de voz que convierta todo el texto a voz que se observe en la pantalla del dispositivo, incluso los diferentes menús que pueda tener la aplicación; un magnificador de pantalla que amplíe lo que se muestra en la pantalla del dispositivo; y hasta algún programa software de ayuda, que facilite el manejo de un menú en un dispositivo. Todos estos productos de apoyo pueden usarse tanto en entorno web con un ordenador de sobremesa, como con distintos dispositivos táctiles.

**Product Owner:** Cliente externo o interno de una organización que define los objetivos del producto, planifica y revisa los objetivos de cada iteración en cada fase de una Metodología Ágil.

**Renderizado Vectorial:** Rasterización. Proceso por el cual una imagen descrita en un formato gráfico vectorial se convierte en un conjunto de píxeles o puntos para ser desplegados en un

medio de salida digital, como la pantalla de una computadora, una impresora electrónica o una imagen de mapa de bits (bitmap).

**Requisito Funcional:** Característica requerida del sistema que expresa una capacidad de acción del mismo, una funcionalidad, generalmente expresada en una declaración en forma verbal.

**Requisito No Funcional:** Característica requerida del sistema, del proceso de desarrollo, del servicio prestado o de cualquier otro aspecto del desarrollo, que señala una restricción del mismo.

**Sandboxing:** Aislamiento de procesos. Mecanismo para ejecutar programas con seguridad y de forma separada.

**Scrum Master:** Líder el equipo que colabora con el cliente (Product Owner) y en con el que en cada reunión se planifica la gestión y el avance del proyecto para mejorar la productividad.

**Service:** Operación de larga duración en segundo plano en Android.

**SSB Bart Group:** Empresa que apoya las iniciativas de accesibilidad de muchas organizaciones mediante el desarrollo de software y soluciones de consultoría, de forma que afronta las necesidades del negocio y técnicas específicas de las organizaciones más afectadas por los requisitos de accesibilidad. Han creado la Plataforma de Gestión de Accesibilidad (AMP), un software que proporciona la infraestructura para gestionar la accesibilidad en todo el ciclo de vida de desarrollo de la empresa.

**SSL:** Transport Layer Security. Protocolo criptográfico que proporciona comunicaciones seguras por una red, como puede ser Internet.

**Tallback:** Lector de pantalla ofrecido por Google en sus dispositivos que lee el elemento que contiene el foco.

**Usabilidad:** Capacidad de un software de ser comprendido, aprendido, usado y ser atractivo para el usuario, en condiciones específicas de uso.

**W3C:** Comunidad internacional donde organizaciones y personas trabajan conjuntamente para desarrollar estándares web.

**WAI:** Web Accessibility Initiative. Iniciativa que desarrolla guías y recomendaciones referentes a los diferentes aspectos de la accesibilidad Web.

**WAI-ARIA:** recomendación internacional que define cómo hacer aplicaciones y contenido web más accesible para personas con discapacidad. Está especialmente enfocado para la generación de contenido dinámico y para interfaces de usuario avanzadas desarrolladas con AJAX, HTML y Javascript. Esta englobada en la Iniciativa para la Accesibilidad Web (WAI) del Consorcio de la Web (W3C).

**WCAG:** Recomendación internacional del consorcio World Wide Web (W3C), sobre cómo hacer accesibles los contenidos de la web a las personas con discapacidad.

**Widget:** Pequeña aplicación o programa, presentado en ficheros pequeños, que dan acceso a funciones frecuentemente usadas en un sistema operativo móvil y proveen de información visual.

## 9. Bibliografía

### 9.1 Bibliografía general

Accesibilidad en Android:

<http://developer.android.com/guide/topics/ui/accessibility/index.html>

[AENOR, 1986]\_Asociación Española de Normalización y Certificaciones la entidad legalmente responsable del desarrollo de las normas técnicas en España. Accesible en:

<http://www.aenor.es/aenor/aenor/perfil/perfil.asp>

[AMP MOBILE] Accesible en <http://www.ssbbartgroup.com/blog/amp-for-mobile-is-here/>

y también en <http://info.ssbbartgroup.com/AMPforMobile.html>

Android Web and Native Accessibility in 4.4 KitKat:

<http://pauljadam.com/kitkata11y/>

Aplicaciones con la API de accesibilidad de Android 4.4:

<https://developer.android.com/guide/topics/ui/accessibility/apps.html>

Aria Live-Regions:

[https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/ARIA\\_Live\\_Regions](https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/ARIA_Live_Regions)

Arquitectura de Android:

<http://androideity.com/2011/07/04/arquitectura-de-android/>

Ayuda contextual en accesibilidad:

<http://olgacarreras.blogspot.com.es/2013/12/ayuda-contextual-de-los-formularios-mas.html>

[BBC, 2005] Accesible en <http://www.bbc.co.uk/guidelines/futuremedia/accessibility/mobile>

[BBC Mobile Accessibility Guidelines] Accesible en

<http://www.bbc.co.uk/guidelines/futuremedia/accessibility/mobile>

Cambios de contexto en accesibilidad:

<http://labda.inf.uc3m.es/awa/en/node/56>

Checklist para accesibilidad de android:



<https://developer.android.com/guide/topics/ui/accessibility/index.html>

Checklist para móviles:

<http://www.digitalgov.gov/2013/07/31/mobile-product-accessibility-testing-resources/>

Conjunto de enlaces de los mejores estándares y best practices (Iheni : Set of Links for Mobile Accessibility):

<http://www.iheni.com/mobile-accessibility-guidelines/>

[CWA 15554, 2006] Base de la certificación europea en Accesibilidad Web. Accesible en <http://www.euracert.org/es/recursos/cwa/>

[DEQUE SYSTEMS, 1999] Accesible en <http://www.deque.com/products/amaze/my-amaze/>

[Deque Guidelines and Resources] Accesible en <https://wiki.state.ma.us/confluence/display/assistivetechologygroup/Accessibility+Guidelines+and+Resources>

[DEQUE SYSTEM AXE] Accesible en "<http://www.deque.com/products/axe/>

[DISCAPNET, 2014] Asociación DISCAPNET. Accesible en

<http://www.discapnet.es/Castellano/areastematicas/Accesibilidad/accesibilidadcomunicacion/Paginas/default1.aspx>

Draft Mobile WCAG Techniques - Mobile Accessibility Task Force:

[https://www.w3.org/WAI/GL/mobile-a11y-tf/wiki/Draft\\_WCAG\\_Techniques](https://www.w3.org/WAI/GL/mobile-a11y-tf/wiki/Draft_WCAG_Techniques)

[eXaminator] Herramienta de evaluación de accesibilidad. Accesible en <http://examinator.ws/>

[Funka Nu, 2000] Accesible en <http://www.funka.com/en/>

[Funka Nu Mobile Accessibility Guidelines]:

<http://www.funkanu.com/en/Our-Expertise/Research-projects/Arkiv-Forskningsrapporter-och-fortroendeuppdrag/Mobile-accessibility-guidelines1/>

Guías de accesibilidad móvil:

<http://olgacarreras.blogspot.com.es/2007/02/web-mvil-y-w3c.html>

Guía para elaborar documentación accesible:

<http://accesibilidadenlaweb.blogspot.com.es/2014/10/guia-para-elaborar-documentacion.html>

<http://fundaciontecsos.es/noticias/guia-para-elaborar-documentacion-digital-accesible>

[HERA] Herramienta de evaluación de accesibilidad. Accesible en <http://www.sidar.org/hera/>



[ISO/IEC 40500, 2012] Estándar que incluye las WCAG 2.0. Accesible en:

[http://www.iso.org/iso/iso\\_catalogue/catalogue\\_tc/catalogue\\_detail.htm?csnumber=58625](http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=58625)

[LEY 11, 2007] Ley de acceso electrónico de los ciudadanos a los Servicios Públicos. Accesible en <https://www.boe.es/boe/dias/2007/06/23/pdfs/A27150-27166.pdf> (septiembre, 2015).

[LEY 27, 2007] Ley que regula la lengua de signos como lengua de las personas sordas, con discapacidad auditiva y sordociegas. Accesible en <http://www.boe.es/boe/dias/2007/10/24/pdfs/A43251-43259.pdf> (septiembre, 2015).

[LEY 56, 2007] Ley de medidas de Impulso de la Sociedad de la Información. Accesible en <http://www.boe.es/boe/dias/2007/12/29/pdfs/A53701-53719.pdf> (septiembre, 2015).

Marco legal de la accesibilidad:

<http://www.iprenafeta.com/2007/11/23/marco-legal-de-la-accesibilidad/>

Mejoras de accesibilidad en Android 4.4:

<http://www.amovil.es/es/blogs/accesibilidad-android-44-kitkat>

Mobile Accessibility: How WCAG 2.0 and Other W3C-WAI Guidelines Apply to Mobile

<http://w3c.github.io/Mobile-A11y-TF-Note/>

Mobile Web Applications Current Status

[http://www.w3.org/standards/techs/mobileapp#w3c\\_all](http://www.w3.org/standards/techs/mobileapp#w3c_all)

Modelo-Vista-Controlador (MVC):

[http://es.wikipedia.org/wiki/Modelo%E2%80%93vista%E2%80%93controlador#Interacci%C3%B3n\\_de\\_los\\_componentes](http://es.wikipedia.org/wiki/Modelo%E2%80%93vista%E2%80%93controlador#Interacci%C3%B3n_de_los_componentes)

[MWABP, 2010] Mobile Web Application Best Practices. Accesible en:

<http://www.w3.org/TR/mwabp/>

[MWBP, 2008] W3C – Mobile Best Practices. Accesible en:

<http://www.w3.org/TR/mobile-bp/>

[MWI, 2005] Iniciativa Web Móvil (MWI) del W3C . Accesible en: <http://www.w3.org/Mobile/>

Normas de accesibilidad para la administración electrónica española:

[http://administracionelectronica.gob.es/pae/Home/pae/Estrategias/pae/Accesibilidad/pae\\_normativa/pae\\_elInclusion\\_Normas\\_Accesibilidad.html#.VZZOrmchw-1s](http://administracionelectronica.gob.es/pae/Home/pae/Estrategias/pae/Accesibilidad/pae_normativa/pae_elInclusion_Normas_Accesibilidad.html#.VZZOrmchw-1s)

Normativas de accesibilidad web:

<http://www.accesibilidadparatodos.com/normativa.html>

Notificaciones dinámicas de deque

<http://www.deque.com/blog/dynamic-notifica>

[Paul J. Adam] Guías de recomendaciones de accesibilidad. Accesible en:

<http://pauljadam.com/favelets/>

Paul J. Adam Demos. Accesible en <http://pauljadam.com/demos/mobile101.html>

Productos de apoyo:

<http://www.observatoriodelaaccsibilidad.es/productos-apoyo/>

[REAL DECRETO 366, 2007] Establece condiciones de accesibilidad y no discriminación de las personas con discapacidad. Accesible en

<https://www.boe.es/boe/dias/2007/03/24/pdfs/A12852-12856.pdf>

[REAL DECRETO 1494, 2007] Aprueba el reglamento sobre las condiciones básicas para el acceso de las personas con discapacidad a las tecnologías, productos y servicios relacionados con la sociedad de la información y medios de comunicación social.

Accesible en <http://www.boe.es/boe/dias/2007/11/21/pdfs/A47567-47572.pdf>

Requisitos de WAI-ARIA:

<http://www.w3.org/TR/WCAG20-TECHS/aria.html>

Responsive Design, buenas y malas practicas, errores comunes:

<http://olgacarreras.blogspot.com.es/2014/01/responsive-design-y-accesibilidad.html>

[SIDAR,2002] Seminario de Iniciativas sobre Discapacidad y Accesibilidad en la Red. Accesible

en <http://www.sidar.org/>

[SSB BART Group, 1997] Organización que provee herramientas de evaluación de accesibilidad.

Accesible en <http://www.ssbbartgroup.com/>.

Standards for Web Applications on Mobile: current state and roadmap - October 2014

<http://www.w3.org/Mobile/mobile-web-app-state/>

Talkback en smartphones:

<http://www.samsung.com/es/article/talkback-en-smartphones-una-ayuda-a-la-discapacidad-visual>

[TAW] Herramienta de evaluación de accesibilidad. Accesible en <http://www.tawdis.net/>

Técnicas HTML para las Pautas de Accesibilidad al Contenido de la Web 1.0

[http://www.discapnet.es/Web\\_accesible/tecnicas/html/WCAG10-HTML-TECHS\\_es.html](http://www.discapnet.es/Web_accesible/tecnicas/html/WCAG10-HTML-TECHS_es.html)

Técnicas de scripting del lado del cliente:

<http://www.w3.org/TR/WCAG20-TECHS/client-side-script.html>

[UNE 139803, 2004] Norma UNE 139803:2004. Requisitos de Accesibilidad para contenidos en la web. Anulada por UNE 139803:2012. Accesible en <http://www.aenor.es/aenor/normas/normas/fichanorma.asp?tipo=N&codigo=N0032576>

[UNE 139803, 2012] Norma UNE 139803:2012. Requisitos de Accesibilidad para contenidos en la web. Anula a UNE 139803:2004. Accesible en <http://www.aenor.es/aenor/normas/normas/fichanorma.asp?tipo=N&codigo=N0049614#.VZZ66fmovRY>

[UNE-EN 301 549, 2014] Norma UNE EN 301 549. Requisitos de Accesibilidad para contenidos en la web. Basada en las WCAG 2.0. Accesible en: <http://www.aenor.es/aenor/actualidad/actualidad/noticias.asp?campo=1&codigo=30377&tipon=1>

[UWEM 1.0, 2006] Metodología de evaluación de las Pautas de Accesibilidad WCAG 1.0. Accesible en <http://www.euracert.org/es/recursos/uwem/>

Validadores de accesibilidad:

[http://www.usableyaccesible.com/recurso\\_misvalidadores.php](http://www.usableyaccesible.com/recurso_misvalidadores.php)

[W3C, 1994] Consorcio World Wide Web. Accesible en <http://www.w3c.es/>

[WAI] Web Accessibility Initiative. Accesible en

<http://www.w3c.es/Traducciones/es/WAI/intro/accessibility/>

[WAI-ARIA, 2014] WAI-ARIA 1.0 Authoring Practices:

<http://www.w3.org/TR/wai-aria-practices>

[WAI-ARIA 1.1, 2014] Recomendación internacional para hacer aplicaciones y contenido web más accesibles. Incluida en la Iniciativa para la Accesibilidad Web (WAI) del Consorcio de la Web (W3C). Accesible en: <http://www.w3.org/TR/2014/WD-wai-aria-1.1-20140612/>

[WAVE] Herramienta de evaluación de accesibilidad. Accesible en <http://wave.webaim.org/>

[WCAG 1.0, 1999] Pautas de Accesibilidad de Contenido Web 1.0. Ampliadas en las WCAG 2.0.  
Accesible en <http://www.w3.org/TR/WCAG10/>

[WCAG 2.0, 2008] Pautas de Accesibilidad de Contenido Web 2.0. Incluye las WCAG 1.0..  
Accesible en: <http://www.w3.org/TR/WCAG20/>

"WCAG 2.0 Should Not Be Applied to Software and Mobile Apps". Artículo.

[Nakata, Ken] Ken Nakata, "WCAG 2.0 Should Not Be Applied to Software and Mobile Apps", 2014. Accesible en:

<http://blog.hisoftware.com/2014/accessibility/wcag-2-0-should-not-be-applied-to-software-and-mobile-apps>

## 9.2 Bibliografía de las herramientas de evaluación de accesibilidad

[AMP MOBILE] Accesible en <http://www.ssbbartgroup.com/blog/amp-for-mobile-is-here/>

y también en <http://info.ssbbartgroup.com/AMPforMobile.html>

[DEQUE SYSTEM AXE] Accesible en "<http://www.deque.com/products/axe/>

[eXaminator] Herramienta de evaluación de accesibilidad. Accesible en <http://examinator.ws/>

[HERA] Herramienta de evaluación de accesibilidad. Accesible en <http://www.sidar.org/hera/>

[TAW] Herramienta de evaluación de accesibilidad. Accesible en <http://www.tawdis.net/>

[WAVE] Herramienta de evaluación de accesibilidad. Accesible en <http://wave.webaim.org/>

## 9.3 Bibliografía del desarrollo y la implementación de la aplicación

Aplicaciones con la API de accesibilidad de Android 4.4:

<https://developer.android.com/guide/topics/ui/accessibility/apps.html>

Aria Live-Regions:

[https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/ARIA\\_Live\\_Regions](https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/ARIA_Live_Regions)

Desarrollo en Android:

<https://developer.android.com/guide/topics/ui/accessibility/index.html>

<https://developer.android.com/reference/android/accessibilityservice/package-summary.html>

<https://developer.android.com/reference/android/view/accessibility/package-summary.html>



<https://developer.android.com/reference/android/view/accessibility/AccessibilityEvent.html>

<https://developer.android.com/reference/android/view/accessibility/AccessibilityEventSource.html>

<https://developer.android.com/reference/android/view/accessibility/AccessibilityManager.html>

<https://developer.android.com/reference/android/view/accessibility/AccessibilityManager.AccessibilityStateChangeListener.html>

<https://developer.android.com/reference/android/view/accessibility/AccessibilityManager.TouchExplorationStateChangeListener.html>

<https://developer.android.com/reference/android/view/accessibility/AccessibilityNodeInfo.html>

<https://developer.android.com/reference/android/view/accessibility/AccessibilityNodeInfo.CollectionInfo.html>

<https://developer.android.com/reference/android/view/accessibility/AccessibilityNodeInfo.CollectionItemInfo.html>

<https://developer.android.com/reference/android/view/accessibility/AccessibilityNodeInfo.RangeInfo.html>

<https://developer.android.com/reference/android/view/accessibility/AccessibilityNodeProvider.html>

<https://developer.android.com/reference/android/view/accessibility/AccessibilityRecord.html>

<https://developer.android.com/reference/android/accessibilityservice/AccessibilityService.html>

<https://developer.android.com/reference/android/accessibilityservice/AccessibilityServiceInfo.html>

<https://developer.android.com/reference/android/view/accessibility/CaptioningManager.html>

<https://developer.android.com/reference/android/view/accessibility/CaptioningManager.CaptioningChangeListener.html>

<https://developer.android.com/reference/android/view/accessibility/CaptioningManager.CaptionStyle.html>

<http://developer.android.com/reference/android/speech/tts/TextToSpeech.html>

Notificaciones dinámicas de deque:

<http://www.deque.com/blog/dynamic-notifica>

Validadores de accesibilidad:



[http://www.usableyaccesible.com/recurso\\_misvalidadores.php](http://www.usableyaccesible.com/recurso_misvalidadores.php)

## Anexo I: Manual de instalación de la aplicación

Para poder ejecutar la aplicación en un dispositivo Android, éste debe disponer como máximo de la versión 4.4 del sistema, e instalar el **ejecutable Android .apk**, con nombre **accesibilidad.apk**. La instalación se podrá realizar con los siguientes pasos:

- 1) Si el archivo lo tenemos en otro dispositivo u en el ordenador y lo queremos usar en el móvil, conectaremos el cable USB del dispositivo para enchufarlo al ordenador, de forma que el sistema solicitará activar el modo de almacenamiento masivo para poder montar la tarjeta SD o reconocer la memoria interna del dispositivo.
- 2) Transferir el archivo APK de la aplicación a la tarjeta SD o memoria interna del dispositivo dónde se quiere ejecutar.
- 3) Indicar a Android que permita la instalación de programas fuera de Google Play. Para ello, hay que desplazarse hasta "**Ajustes>Seguridad>Orígenes desconocidos**" y activar la opción.
- 4) Al finalizar este proceso ya se puede instalar la aplicación. Abrimos el administrador de archivos que queramos y se busca el archivo APK en la carpeta creada en la tarjeta SD o en la memoria interna y lo ejecutamos para lanzar la aplicación.

## Anexo II: Manual de usuario de la aplicación

Al lanzar la aplicación lo primero que observa el usuario es la `PrincipalActivity` de la aplicación, la pantalla principal de la misma. Para analizar una aplicación nativa deberá seguir los siguientes pasos:

- 1) Abrir con el explorador de archivos y buscar la aplicación que se va analizar (el ejecutable APK).
- 2) Seleccionar ese archivo APK y pulsar en el botón de análisis de la aplicación.
- 3) Esperar a que termine el análisis y se muestre un informe con los resultados obtenidos.
- 4) Una vez obtenido el informe, el usuario podrá volver a la pantalla inicial para realizar otro análisis.
- 5) Para salir de la aplicación, el usuario deberá dejar de usarla, y al estar en segundo plano, la aplicación se cerrará a los 5 minutos que no se detecte actividad.
- 6) Si ocurre algún error, sea al abrir un archivo o durante la ejecución de la aplicación, ésta se redirige hacia una pantalla que muestra el tipo de error, y se detiene la aplicación.

### Anexo III: Evolución de accesibilidad en Android

Distribución	Funcionalidad accesible	Descripción/Ventajas
Android Beta	No existen características de accesibilidad.	Sistema Android muy reciente para centrarse en accesibilidad.
Android 1.0 Apple Pie	No existen características de accesibilidad.	Sistema Android muy reciente para centrarse en accesibilidad.
Android 1.1 Banana Bread	No existen características de accesibilidad.	Sistema Android muy reciente para centrarse en accesibilidad.
Android 1.5 Cupcake	Clase de Java TextToSpeech.	Sintetiza texto para su reproducción inmediata o crea un archivo de sonido (voz).
Android 1.6 Donut	Inclusión de una API de accesibilidad.	Lector de pantalla y de texto a voz para ciegos. Todas las partes de la interfaz de usuario son accesibles a través de la rueda de desplazamiento y los controles de imagen tiene una asociación textual. Aplicaciones que proporcionan feedback hablado (Talkback), auditivo (Soundback) y háptico (Kickback).
Android 2.0/2.1 Eclair	Ligeras mejoras en accesibilidad.	Optimización de las características de accesibilidad ya existentes.
Android 2.2 Froyo	Búsqueda de voz en Google y en contactos.	Se puede instalar otro motor de voz y cambiar la velocidad de la voz.
Android 2.3 Gingerbread	Ligeras mejoras en accesibilidad.	Correo de Gmail y calendario de Android accesibles.
Android 3.0 Honeycomb	Se incorporan scripts para la clase Java WebView.	Opción de accesibilidad para introducir texto en Android mediante el teclado Eyes-Free Keyboard.

Android 4.0 Ice Cream Sandwich	Tamaño de fuente ajustable e incorporación de la exploración táctil Explore By Touch en Talkback.	El sistema puede generar eventos de accesibilidad en función de cómo se interaccione con la pantalla del dispositivo, según se pulse con el dedo. El lector de pantalla puede sintetizar y reproducir el contenido y la descripción del elemento por el que se esté navegando en ese momento.
Android 4.1 Jelly Bean	Inclusión de Explore by Touch y BrailleBack.	Explore by Touch permite explorar por le tacto, arrastrando lentamente el dedo por la pantalla para mover el foco. Se escuchan los iconos, botones y otros objetos en pantalla. Si se pulsa dos veces en cualquier lugar de la pantalla se activa el elemento enfocado.  BrailleBak permite conectar una pantalla Braille actualizable al dispositivo Android mediante Bluetooth. Si se utiliza con el lector de pantalla TalkBck, permite un uso combinado de voz y Braille.
Android 4.2 Jelly Bean	Inclusión del magnificador de pantalla.	Se puede ampliar cualquier contenido o zona de la pantalla.
Android 4.3 Jelly Bean	Compatibilidad del lector con el magnificador de pantalla.	Permite ampliar una imagen de la pantalla, junto con la síntesis de voz del lector de pantalla TalkBack.
Android 4.4 Kit Kat	Inclusión de un apartado de preferencias para subtítulos.	El usuario puede configurar idioma, tamaño y estilo de texto. Las aplicaciones pueden acceder a esa configuración y ajustar su presentación para satisfacer las preferencias del usuario.
Android 5.0 Lollipop	Mejorado el soporte para personas daltónicas y de baja visión  Asistente de búsqueda por voz Ok Google.	Mayor contraste o inversión de colores para mejorar la legibilidad.  Ajuste de pantalla para mejorar la diferenciación.  Con el asistente de voz Ok Google se consigue un fácil acceso a información y realización de tareas, incluso mientras la pantalla del dispositivo móvil está apagada o el usuario está en movimiento.

Android 5.1 Lollipop	Mejoras en el asistente de búsqueda por voz Ok Google.	Optimización de Ok Google.
Android 6.0 Marshmallow	Google Now On Tap Otras mejoras todavía en desarrollo.	Al recibir un correo con información sobre un servicio o producto, Google Now On Tap permite, manteniendo pulsado el dedo en el botón Home, obtener información sobre ese producto o servicio.

Tabla 15: Evolución de la accesibilidad en las distintas distribuciones de Android